

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Profilier les étudiants pour leur donner envie d'apprendre

Kazadi Ntambwe, Hervé

*Award date:*  
2021

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Université de Namur  
Faculté d'informatique  
Année académique 2020–2021

**Profilier les étudiants pour leur  
donner envie d'apprendre**

Hervé KAZADI NTAMBWE



Promoteur : \_\_\_\_\_ (Signature pour approbation du dépôt - REE art. 40)  
Bruno DUMAS

Co-promotrice : Julie HENRY

Mémoire présenté en vue de l'obtention du grade de  
Master en Sciences Informatiques.

## I. Remerciements

Je tiens à remercier ma promotrice, Mme Julie HENRY, assistante à la faculté d'informatique à l'université de Namur, et mon promoteur, Mr. Bruno DUMAS, Professeur à la faculté d'informatique, pour leurs précieux conseils et leur assistance durant toute la préparation de ce mémoire.

Je remercie également mon amie, Mme Pascaline MUGISHO, pour ses conseils en matière des notions statistiques.

Un remerciement spécial à mon épouse Magali pour son soutien indéfectible, sa patience et ses encouragements. Je n'oublie pas de remercier mes quatre enfants Serena, Roni, Allegrah et Gracia pour leur compréhension et leur amour.

Un grand merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Enfin, je remercie mon Dieu pour la bonne santé et la force.

## II. Résumé

Tous les étudiants inscrits en première année dans la faculté Ingénieur de Gestion (ingénierie en management de l'information – INGMI) ou dans la faculté informatique (INFO) à l'université de Namur, sont confrontés à un cours d'introduction à la programmation (INFOB131). Cependant, il a été constaté que ces étudiants n'arrivent pas tous avec les mêmes compétences pour réussir ce cours. Cette étude extrait les compétences des étudiants du questionnaire qui leur avait soumis avant le début de l'année académique afin de répondre à la question de savoir quels sont les facteurs qui influencent la réussite ou pas. Ce qui a permis de montrer, malgré un faible nombre d'étudiants, qu'une autodétermination élevée et le fait d'être une fille favorisent la réussite en INGMI et quant à la population INFO, la participation aux cours préparatoires combinée avec le fait d'avoir eu des cours d'informatique en secondaire ainsi que l'apprentissage seul ou pas d'au moins un langage de programmation influencent la réussite au cours de programmation.

Mots-clés : informatique, programmation, introduction à la programmation, pensée computationnelle, facteurs de réussite, bagage, régression logistique, arbre de décision.

### III. Table des Matières

I.	Remerciements.....	1
II.	Résumé.....	2
III.	Table des Matières .....	3
IV.	Introduction.....	5
	IV.1. Objectif .....	5
	IV.2. Contexte .....	5
	IV.3. Méthode de travail .....	6
V.	État de l’art.....	7
	V.1. Programmation vs Pensée Computationnelle.....	7
	V.1.1. La programmation informatique .....	7
	V.1.2. La pensée computationnelle .....	8
	V.1.3. Lien entre la pensée computationnelle et la Programmation.....	9
	V.2. Réussites et échecs au cours de programmation .....	9
	V.2.1. Aperçu de la situation .....	9
	V.2.2. Les facteurs de réussite et d’échec.....	10
	V.2.3. Réduire l’échec.....	12
	V.3. L’informatique dans l’enseignement secondaire .....	14
	V.3.1. Aperçu.....	14
	V.3.2. L’informatique en Belgique francophone en fin de secondaire .....	14
	V.4. Les motivations des choix des étudiants .....	17
	V.4.1. L’autodétermination .....	17
	V.4.2. Engagement vers un but. ....	19
	V.4.3. Quelles sont ces motivations .....	21
VI.	Développement de la recherche .....	23

VI.1. La Recherche .....	23
VI.1.1. Rappel du contexte .....	23
VI.1.2. Question de recherche.....	23
VI.1.3. Hypothèses.....	23
VI.2. Les données .....	24
VI.2.1. Enquête et Population .....	24
VI.2.2. La base de données .....	25
VI.2.3. Traitement des données.....	28
VI.2.4. Répartition des étudiants.....	34
VII. Analyse et Résultats.....	45
VII.1. Choix du modèle .....	45
VII.2. Méthodes d'analyse .....	46
VII.2.1. Analyse Bivariée .....	47
VII.2.2. Analyse multivariée.....	51
VII.3. Profil des étudiants en INGMI.....	69
VII.3.1. Rappel des résultats.....	69
VII.3.2. Profil .....	71
VII.3. Profil des étudiants en INFO .....	71
VII.3.1. Rappel des résultats.....	71
VII.3.2. Profil .....	73
VIII. Discussion .....	74
VIII.1. Biais.....	76
IX. Conclusion .....	78
X. Bibliographie .....	81
XI. Annexe .....	86
XI.1. Questionnaire soumis aux étudiants .....	86

## IV. Introduction

### IV.1. Objectif

L'objectif de ce mémoire est de profiler les étudiants de la faculté Ingénieur de Gestion (ingénierie en management de l'information – INGMI) et les étudiants de la faculté informatique (INFO) de l'université de Namur en déterminant les facteurs qui influencent leur réussite ou leur échec au cours d'introduction à la programmation dispensé en première année. Cela veut dire qu'un profil type d'étudiants INGMI sera présenté et un autre d'étudiants d'INFO.

Chaque profil sera composé de compétences acquises par les étudiants avant leur entrée à l'université, et qui influencent leur réussite ou leur échec au cours d'introduction à la programmation. Ces compétences ou ces qualités qui constituent des *bagages*, ont été séparées en quatre groupes : bagage en informatique acquis durant les études secondaires, bagage en programmation, connaissances en langage Python, bagage lié à la motivation.

Les profils de réussite ou d'échec des étudiants d'INGMI et d'INFO pourraient permettre à l'université de trouver des solutions pour aider les futurs nouveaux étudiants dont le profil ne serait pas susceptible de conduire à la réussite au cours d'introduction à la programmation.

### IV.2. Contexte

Selon « l'annuaire des formations<sup>1</sup> », le cours d'introduction à la programmation (INFOB131) donné aux étudiants de la faculté Ingénieur de Gestion (ingénierie en management de l'information – INGMI) et de la faculté informatique (INFO) à l'université de Namur a pour but d'initier l'étudiant à la programmation, afin qu'il puisse résoudre des problèmes simples à l'aide d'algorithmes adaptés. Plusieurs notions de programmation sont traitées : les variables, les fonctions, la spécification, les boucles, les structures de données, les bases d'algorithmique, etc. Ces notions sont accompagnées des séances d'exercices permettant aussi à l'étudiant de découvrir le langage Python.

Les étudiants sont évalués premièrement par un examen écrit portant sur la théorie et les exercices et deuxièmement sur leurs capacités à mettre en œuvre les compétences et connaissances du cours.

---

<sup>1</sup> Lien vers l'annuaire des formations : <https://directory.unamur.be/teaching/courses/INFOB131/2020>

Le cours d'introduction à la programmation est considéré en INFO comme un cours de dix crédits (60h+45h), alors qu'en INGMI, il ne compte que neuf crédits. Quelle que soit la faculté, les cohortes d'étudiants assistent à un même cours qui se déroule au premier quadrimestre, c'est-à-dire de septembre à décembre, avec l'examen organisé en janvier.

Il a été constaté que les étudiants inscrits au cours INFOB131 ne possèdent pas tous les mêmes compétences en programmation. Dans le souci de mettre en évidence, pour chaque étudiant, l'existence (ou l'absence) d'un bagage en programmation en début d'année académique, un questionnaire a été soumis à ces étudiants : les étudiants de la faculté Ingénieur de Gestion (ingénierie en management de l'information – INGMI) et les étudiants de la faculté informatique (INFO). Il est évident que certains étudiants arrivent avec des compétences développées qui devraient normalement faciliter leur réussite dans un cours de programmation alors que d'autres n'ont pas eu l'opportunité de développer de telles compétences.

### IV.3. Méthode de travail

La méthode de travail consistait d'abord à rassembler toutes les informations nécessaires découlant des réponses données par les étudiants lors de l'enquête. Ensuite, à déterminer pour chaque réponse, le bagage auquel elle fait partie. Enfin à traiter les différentes variables afin qu'elles soient idéales pour l'analyse.

Les variables traitées devaient être analysées afin d'obtenir des éléments de réponses vis-à-vis de l'objectif fixé. Deux modèles ont été utilisés afin de déterminer les variables influençant la note des étudiants : la régression logistique et l'arbre de décision.

Ces analyses ont été faites en utilisant le langage de programmation R via le logiciel libre « *R Studio* » couramment utilisé en statistiques et en sciences des données. L'arbre de décision a été réalisé au moyen de l'outil « *Orange Data Mining* », logiciel libre d'exploration des données et de modélisation visuelle, utilisant le langage Python.



## V. État de l'art

Le but d'un cours de programmation est, entre autres, de développer chez les étudiants les compétences de résolution de problème. Cette aptitude est notamment liée au concept de pensée computationnelle (ou pensée informatique). Si ce mode de pensée est considéré, partout dans le monde, comme un incontournable à enseigner aux enfants dès le plus jeune âge comme ce serait le cas au Royaume-Uni où les enfants de 5 à 7 ans seraient déjà initiés à la compréhension de ce que sont les algorithmes d'après [Watson, C. et Li, C., 2014], la programmation génère, pour sa part, plus de réticence. Pour cause, l'apprentissage de la programmation est considéré comme difficile d'après [Gomez et al, 2007]. [B. Hanks et al, 2004] ont constaté qu'au niveau universitaire, beaucoup d'étudiants ont du mal à apprendre avec ce cours. [Nadigam et Bathula, 2011], quant à eux, constatent que beaucoup d'étudiants rencontrent des difficultés pour réussir leur apprentissage de la programmation. De nombreuses études abordent cette problématique.

Qu'est-ce qui distingue la pensée computationnelle, accessible aussi aux enfants, et la programmation ? Quelles sont les difficultés rencontrées dans l'apprentissage de la programmation ? Qu'est-ce qui pourrait aider à surmonter ces difficultés ?

En apportant des réponses à ces questions, cet état de l'art pose les briques de base sur lesquelles va s'appuyer la recherche.

### V.1. Programmation vs Pensée Computationnelle

#### V.1.1. La programmation informatique

Plusieurs définitions de la programmation informatique ont été suggérées. Les auteurs [Adu-Manu et al, 2013] ont défini la programmation comme « *un art qui requiert des aptitudes individuelles pour transformer les problèmes, les défis en solutions* ». [Kirsti Ala-Mutka], lui, a écrit que « *l'art de programmer inclut : la connaissance des outils et des langages de programmations, les compétences pour résoudre les problèmes et les stratégies efficaces pour la conception et l'implémentation d'un programme* ». La Professeure [Magdalena Andrzejewska, 2018]) rappelle que « *Programmation ne veut pas seulement dire une aptitude à coder liée à une connaissance de syntaxe d'un langage choisi et des opérations telles que la modification d'un code, sa compilation et son exécution* », mais que « *c'est un ensemble de compétences efficaces mises ensemble pour la résolution des problèmes* ». Sur base de ces définitions, il apparaît évident que « *parler de la programmation* », c'est aussi *parler de la*

« *résolution des problèmes donnés* » à savoir « *trouver des solutions à un problème* » en utilisant une syntaxe et une sémantique propres à un langage de programmation.

Mais avant de résoudre un problème, celui-ci devrait être d'abord compris, comme la façon dont il doit être résolu [Henry, J & Joris, N, 2016]. Pour pouvoir y arriver, il est important que l'étudiant dispose de certaines compétences. Plusieurs études ont tenté d'identifier les atouts à avoir pour réussir en programmation :

- Être bon en mathématiques [Patil et Goje, 2009] [Ronan, 2005]
- Avoir les capacités en logique [Nowaczyk, R., 2019]
- Avoir la capacité de résoudre des problèmes [Nowaczyk, R., 2019]
- Avoir une expérience antérieure en programmation [Bergin, S & Reilly, R., 2005]
- Être motivé [C. Bereiter and E. Ng, 1991]

#### V.1.2. La pensée computationnelle

Une des qualités indéniables pour réussir à programmer serait d'être doté de la pensée computationnelle.

[Selby et Woollard, 2013] définissent la pensée computationnelle comme étant « *un processus de la pensée composé de cinq concepts principaux : l'abstraction, la décomposition, la pensée algorithmique, l'évaluation et la généralisation* ». [Wing J., 2011] la décrit comme « *un processus de la pensée impliqué dans la formulation des problèmes et de leurs solutions de telle sorte que les solutions soient représentées dans une forme dans laquelle elles peuvent être réalisées, concrétisées par un agent de traitement d'information* ». Cette dernière définition a été simplifiée par [Aho, 2012], la pensée computationnelle est alors vue comme « *des processus de la pensée impliqués dans la formulation des problèmes de telle sorte que leurs solutions soient représentées comme des algorithmes* ». Enfin, la pensée computationnelle est décrite à travers quatre concepts<sup>2</sup> et ressemblant fortement à ceux présents dans la définition de Selby et Woodlard (2013) :

- **Décomposition** : Diviser des problèmes complexes en des problèmes plus petits
- **Reconnaissance des modèles** : Faire un lien avec des problèmes similaires déjà connus
- **Abstraction** : Ressortir que ce qui est essentiel et ignorer ce qui n'est pas pertinent.
- **Algorithme** : Concevoir des étapes simples pour résoudre les problèmes.

Plusieurs articles discutant ce qui doit être enseigné dans l'enseignement obligatoire, préconisent l'étude de ce processus qu'est la pensée computationnelle [Quinn, H. et al, 2012]

---

<sup>2</sup> Source : <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>

[Wilensky, U et al, 2014] [Wing J. E., 2011] et un autre affirme que durant ces années 2020, qu'un « *job sur deux serait un job dans le domaine informatique* » [Kaczmarczyk, L., & Dopplick, R. (2014)] d'où l'importance de savoir ce qu'est la pensée computationnelle.

### V.1.3. Lien entre la pensée computationnelle et la Programmation

Il est évident, au regard des définitions proposées ci – dessus de la pensée computationnelle et de la programmation informatique, qu'il existe bel et bien un lien entre ces deux notions. Une question se pose alors : Est-ce que posséder la *pensée computationnelle (et les concepts qui la définissent)*, *garantit d'être bon en programmation* ? Avant de répondre à cette question, il est essentiel de préciser que, bien qu'elle soit souvent associée à la programmation, la pensée computationnelle peut être apprise sans l'utilisation d'un ordinateur [Thorson K., 2018].

Ceci dit, *parler de la programmation*, ce serait parler aussi de la résolution des problèmes et la pensée computationnelle aiderait dans ce sens car aux vues de ses caractéristiques mentionnées ci-haut, elle permet d'avoir une approche sur comment résoudre n'importe quel problème, y compris lié à l'informatique. Il y aurait donc une relation mutuelle entre la pensée computationnelle et la programmation résumée par cette phrase : « *Savoir Programmer démontrerait que l'on est doté des compétences computationnelles* » [Grover, Shuchi & Pea, 2013]. Autrement dit, celui qui est doté de la pensée computationnelle, devrait être capable de programmer. D'un autre côté « *la programmation permettrait d'améliorer sa pensée computationnelle* » [Armoni et al, 2015] et « *les concepts utilisés dans la programmation peuvent être reconnus comme bases de la pensée computationnelle* » [Brennan, K. & Resnick, M., 2012]. Il existerait donc bel et bien un impact mutuel et une influence importante entre la programmation et la pensée computationnelle. De ce fait, la pensée computationnelle, qui est, rappelons-le, une manière de penser, constituerait donc un atout majeur pour savoir mieux programmer.

## V.2. Réussites et échecs au cours de programmation

### V.2.1. Aperçu de la situation

La programmation est un cours central dans la filière informatique. De plus, de nos jours, la programmation prend une place de plus en plus importante dans l'évolution du monde. La demande des programmeurs expérimentés est à la hausse à l'échelle globale [Watson, C. et

Li, C., 2014], d'où l'importance, pour des étudiants en filière informatique, de réussir dans cette matière qu'est la programmation.

Mais cela ne semble pas facile pour tout le monde. Beaucoup d'articles discutent de l'échec des étudiants au cours de programmation au début de leur parcours académique. Cela démontre que ce sujet inquiète et qu'il nécessite une attention particulière afin de trouver des solutions. Il semble exister des facteurs favorisant la réussite en programmation et d'autres, l'échec. Parmi ces facteurs, il y a :

- Le bagage (background)
- La motivation
- Les capacités cognitives
- Les méthodes d'apprentissage

### V.2.2. Les facteurs de réussite et d'échec

#### 1. Bagage de l'étudiant

Le bagage de l'étudiant fait référence au passé de l'étudiant avant le début de son année académique, à ce qu'il a appris durant son parcours dans l'enseignement obligatoire, notamment l'enseignement secondaire. Deux aspects importants ont été retenus :

- **Le bagage en Maths** : Plusieurs articles ont conclu que les étudiants ayant suivi les options sciences et mathématiques au secondaire ont plus de chance de réussir [Bergin, S. et Reilly, R., 2005] [Patil et Goje, 2009] [Ronan, 2005] [Wilson, B.C. et Shrock, S., 2004]. Les mathématiques seraient donc un prédicteur de la performance en programmation.
- **Le bagage en Programmation** : Un étudiant ayant déjà eu des cours de programmation au secondaire aurait un avantage par rapport aux autres étudiants [Hagan, D. and Markham, S. 2004] [Wiedenbeck, S. et al. 2004] [Holden, E. and Weeden, E. 2004]

D'autres aspects n'ont pas été considérés. Ainsi, concernant le **bagage en informatique**, il est apparu qu'un étudiant qui ne maîtrise pas l'informatique ne finit pas spécialement en échec [Dunican, E., 2002]. Dès lors, le manque de connaissances en informatique n'a pas été considéré comme un facteur amenant l'échec.

#### 2. La motivation de l'étudiant

La motivation est l'expression du but poursuivi par l'étudiant en décidant de suivre une filière informatique dans son parcours académique. C'est aussi l'engagement qu'a l'étudiant

dans son apprentissage des matières qui composent cette filière, dont l'introduction à la programmation. Le manque de motivation ou une motivation insuffisante serait un réel facteur d'échec et d'abandon. Ainsi :

- ***Le manque d'autodétermination*** rendrait la réussite de l'étudiant plus laborieuse [C. Bereiter and E. Ng. 1991] ;
- ***Le manque de motivation*** : Les étudiants qui ne sont pas du tout motivés auraient du mal dans l'apprentissage de la programmation et seraient susceptibles d'échouer [T. Jenkins, 2002] ;
- ***Le manque d'engagement*** : Un étudiant moins engagé envers le but fixé aurait moins l'intention de persévérer dans ce but [Boudrenghien et al, 2013], le but ici étant bien sûr de réussir au cours d'introduction à la programmation.

### 3. Les capacités cognitives de l'étudiant

Tout le monde n'est pas appelé à devenir informaticien. Tout le monde n'a pas les capacités de faire le métier de programmeurs. En terme des capacités cognitives, il faut souligner [M. N. Ismail et al, 2010] [M. Butler and M. Morgan, 2007] :

- ***Le manque de capacités de raisonnement logique*** : L'étudiant n'arrive pas à savoir par où commencer, à savoir ce dont il a besoin pour arriver à la solution. Il n'arrive à traduire un problème textuel en une suite d'instructions qui l'emmènera à une solution.
- ***Le manque de compétences dans la résolution des problèmes*** : Cela est souvent dû à une non-compréhension du problème posé et à un manque d'analogie aux problèmes déjà résolus.

Ce sont donc les manques liés à la pensée computationnelle. Ces manques peuvent être comblés pour certains et pour d'autres, non. Pour ces derniers, il leur faudra une motivation supplémentaire.

### 4. Les méthodes d'apprentissage de l'étudiant

Il s'agit de considérer comment les étudiants s'y prennent pour acquérir les compétences, les connaissances en programmation. Pas besoin de démontrer qu'un mauvais apprentissage d'un cours, quel qu'il soit, conduirait à un échec. Sont considérés :

- ***Les méthodes d'étude*** [Gomes, A. & Mendes, A., 2007] : Les étudiants qui apprennent la programmation en mémorisant, par exemple, les syntaxes du langage de programmation utilisé comme s'il mémorisait des formules utilisées dans d'autres

matières risquent de ne pas être capables de trouver une solution à un problème dans un cours de programmation.

- **Le temps de travail** [Gomes, A. & Mendes, A., 2007] : Les étudiants qui n'accordent pas assez de temps de travail ou pour s'exercer en programmation auront du mal à acquérir les compétences en programmation.
- **La persistance** [Gomes, A. & Mendes, A., 2007] : les étudiants qui abandonnent souvent un problème lorsqu'ils n'arrivent pas à trouver une solution aussi rapidement qu'ils le souhaiteraient, auraient des difficultés dans leur apprentissage.

### V.2.3. Réduire l'échec

L'échec peut être défini comme étant le fait de ne pas atteindre un but tout simplement. Le but de chaque étudiant inscrit dans un cursus étant de réussir, si un étudiant n'atteint ce but, c'est un échec. De façon plus formelle, concernant le milieu universitaire, l'échec dans un cours revient à obtenir moins de points que la moyenne établie pour la matière.

#### 1. Pourquoi devons-nous réduire l'échec

La programmation devient de plus en plus incontournable dans le monde dans lequel on vit et dans lequel il y a un besoin de plus en plus d'informaticiens [US Bureau of Labor Statistics, 2013] et plus particulièrement des développeurs et des programmeurs. Selon le COO (Chief Operating Officer – Directeur opérationnel) du cabinet de recrutement Hudson, Mr MEYSMAN, les informaticiens sont parmi les trois profils les plus recherchés au niveau des cadres [Sury, C., 2020].

Il est donc important de réduire l'échec afin de voir le maximum d'étudiants inscrits dans une filière informatique terminer leur parcours universitaire et apporter leurs compétences pour contribuer à cette montée du numérique. Réduire l'échec est également important pour l'étudiant lui-même. En effet, selon [Galand et al, 2005], un échec peut amener un étudiant à perdre confiance en ses capacités. De plus, un étudiant qui échoue une première fois à l'université a plus de risque d'échouer par après dans son parcours universitaire et tout cela a un coût.

#### 2. Comment réduire l'échec

Réduire l'échec correspond à remédier aux facteurs qui causent cet échec. Pour tous ces facteurs cités précédemment, des solutions ont été proposées afin de permettre aux étudiants,

susceptibles d'échouer au cours de leur apprentissage de la programmation, de s'ajuster afin de changer leur échec annoncé en réussite attestée. La liste des solutions proposée ci-dessous n'est pas exhaustive.

### **Le bagage de compétences**

#### *Les Maths*

- [Gomes, A. & Mendes, A., 2007] préconisent de mettre les étudiants démontrant des faiblesses en Maths dans un environnement qui implicitement ou explicitement les mettront face aux concepts mathématiques qui pourraient être utiles pour des problèmes de programmation typiques.

#### *La programmation*

- Pour les étudiants n'ayant pas de bagage en programmation, il faudra veiller à identifier ceux qui auraient besoin d'un soutien supplémentaire. Il s'agit également de privilégier le travail en groupe, des laboratoires, etc. [Gomes, A. & Mendes, A., 2007]

### **La motivation**

#### *Autodétermination et Engagement envers un but*

- Pour les étudiants dont la motivation exprimée indiquerait une autodétermination insuffisante ou un engagement insuffisant, il s'agit de les amener à découvrir l'impact de la programmation dans la société et dans la vie quotidienne en proposant des activités de résolution des problèmes simples par des jeux [Gomes, A. & Mendes, A., 2007].

### **Les capacités cognitives de l'étudiant**

#### *Résolution des problèmes*

- Si un problème est difficile, il convient de faire une analogie avec des problèmes traités précédemment [Gomes, A. & Mendes, A., 2007].
- Amener l'étudiant à s'autocritiquer et à parfaire sa solution en l'amenant à défendre sa réponse face aux questions de ses camarades [Gomes, A. & Mendes, A., 2007].

### **Les méthodes d'apprentissage**

#### *Les méthodes d'étude, le temps de travail, la persistance*

- Il faut donner la possibilité aux étudiants de beaucoup pratiquer [Gomes, A. & Mendes, A., 2007].

- Parce que ces facteurs sont liés à la motivation, il faut privilégier les projets en groupe pour amener l'étudiant à travailler. Une étude [Adu-Manu et al, 2013] a montré que parmi les étudiants sondés, 74% trouveraient que les méthodes d'enseignement par projets les aident dans la compréhension des concepts et rendraient plus claires leurs compétences
- Il est essentiel d'intégrer des mécanismes qui vont empêcher les étudiants d'abandonner leur recherche de solution [Gomes, A. & Mendes, A., 2007].

### V.3. L'informatique dans l'enseignement secondaire

#### V.3.1. Aperçu

Dans leur article, [Dupriez & Vandenberghe, 2003] ont montré que *« l'échec et la réussite dans les études universitaires étaient aussi liés à ce qu'on apprend à l'école secondaire »*. Cela montre l'importance et l'impact des connaissances acquises à l'école secondaire dans la réussite des étudiants à l'université. Cela pourrait être rapporté à la programmation en particulier : *« l'échec et la réussite dans le cours d'introduction à la programmation à l'université seraient liés à ce qu'on apprend comme cours d'informatique à l'école secondaire »*.

Bien qu'il a été dit que le bagage en informatique ne serait pas prédicteur de la bonne performance en programmation à l'université [Dunican, E. 2002], ça ne signifie pas qu'il n'apporterait rien pour autant. Il est important d'étudier cet aspect d'une autre manière en se concentrant sur les caractéristiques de la pensée computationnelle (décomposition, reconnaissance des modèles, abstraction et algorithme). Il s'agit d'identifier quelles matières faisant partie du bagage en informatique influenceraient la performance des étudiants.

#### V.3.2. L'informatique en Belgique francophone en fin de secondaire

Un coup d'œil est porté au programme d'études « informatique » proposé pour le troisième degré en Belgique francophone [Administration générale de l'enseignement, 2016]. L'un des objectifs de la formation informatique, selon ce programme, vise à *« créer et à développer l'esprit de rigueur, d'organisation, de précision et de qualité »* et permet aussi à ces étudiants *« d'apprendre à construire des solutions techniques pour répondre à des besoins humains »*.

Ce programme dresse la liste des cours informatiques donnés dans l'enseignement en fin de secondaire en Belgique francophone. Pour chacun des cours, les compétences que sont



censés acquérir les élèves sont énumérées. Ces compétences peuvent donc être une source d'information pour évaluer si un cours comprend des éléments pouvant être considérés comme comprenant au moins un des concepts de la pensée computationnelle.

En effet, [Grover, Shuchi & Pea, Roy. 2013], dans leur travail, ont listé des éléments proposés et « largement acceptés » par différents chercheurs et enseignants en informatique, comme incluant la pensée computationnelle :

- Abstractions et généralisation des modèles (1)
- Traitement systématique des informations (2)
- Systèmes des symboles et des représentations (3)
- Notions algorithmiques des flux de contrôle (4)
- Décomposition des problèmes structurés (modularisation) (5)
- Pensée itérative, récursive et parallèle (6)
- Logique conditionnelle (7)
- Contraintes d'efficacité et de performance (8)
- Débogage et détection systématique des erreurs. (9)

Voici un aperçu des compétences listées dans le programme d'études informatique en Belgique francophone :

### **Le codage de l'information (codage binaire, codage de textes, d'images, etc.)**

- Convertir d'une base à l'autre
- Appliquer des principes arithmétiques et logiques sur des nombres binaires
- Traiter des images optimisées pour des besoins spécifiques
- Traiter des séquences sonores optimisées pour des besoins spécifiques
- Traiter des séquences animées (vidéos, animation intégrée, ...).
- Mettre en forme un document long dans un logiciel de traitement de texte
- Traiter des données à l'aide d'un logiciel tableur
- Produire une présentation à l'aide d'un logiciel de présentation assistée par ordinateur

### **L'architecture de base d'un ordinateur**

- Établir ou choisir parmi un ensemble d'offres, la configuration qui répond le mieux à un besoin précis
- S'assurer de l'absence de risques avant, pendant et après toute manipulation du matériel

### **Le système d'exploitation (fonctions et rôles) et la gestion de fichiers**

- Appliquer l'emploi des unités informatiques à des situations concrètes
- Maîtriser l'arborescence préexistante du système d'exploitation
- Utiliser l'interface du système d'exploitation et les périphériques
- Maîtriser des invariants fonctionnels de logiciels

### **Les réseaux et Internet**

- Concevoir un réseau informatique répondant à des besoins spécifiques
- Sécuriser un réseau TCP IP
- Partager les informations et les ressources de manière sécurisée.

### **La création de page Web (langage HTML)**

- Concevoir et réaliser un site web dynamique, ergonomique et esthétique

### **La programmation et réalisation d'algorithmes**

- Analyser une situation problème et modéliser la solution sous forme d'algorithme
- Traduire un algorithme dans un langage informatique structuré
- Déclarer les variables d'exécution du programme
- Écrire une séquence en appliquant les instructions d'entrée et de sortie
- Traduire une opération, une fonction en langage de programmation
- Écrire un programme utilisant des séquences alternatives, répétitives
- Construire un programme modulaire
- Manipuler les tableaux
- Manipuler les structures d'enregistrements

### **La création des bases des données**

- Modéliser une base de données à partir d'une situation concrète
- Exploiter une base des données

Ci-dessous, le lien entre les cours et leurs compétences de pensée computationnelle avec un des éléments de la pensée computationnelle proposés par [Grover, Shuchi & Pea, Roy. 2013].

Tableau V.1. Cours et compétences de pensée computationnelle

Cours	Compétences de pensée computationnelle
Codage de l'information	Convertir d'une base à l'autre (3) Appliquer des principes arithmétiques et logiques sur des nombres binaires. (3) Traiter des images optimisées pour des besoins spécifiques (2) Traiter des séquences sonores optimisées pour des besoins spécifiques (8) Traiter des séquences animées (vidéos, animation intégrée, ...) (8)
Architecture de base d'un ordinateur	Établir ou choisir parmi un ensemble d'offres, la configuration qui répond le mieux à un besoin précis (8)
Cours de réseau	Concevoir un réseau informatique répondant à des besoins spécifiques (5) Sécuriser un réseau TCP IP (5) Partager les informations et les ressources de manière sécurisée (2)
Programmation et réalisation d'Algorithme	Toutes les compétences (4)
Bases de données	Modéliser une base de données à partir d'une situation concrète (5)

#### V.4. Les motivations des choix des étudiants

Choisir une filière, à la sortie de l'enseignement secondaire, est quelque chose de très important dans la vie de nombreux nouveaux étudiants, et est déterminant pour leur avenir. Tout choix est toujours accompagné de certaines raisons et de certains critères définis par celui qui choisit. La motivation est une forme de détermination et d'engagement personnel dans l'accomplissement d'une tâche, dans le but d'atteindre un objectif fixé. Les élèves qui sortent donc de l'enseignement secondaire en choisissant de s'inscrire dans une filière informatique, ont eux aussi des raisons qui les ont poussés à faire ce choix, ils ont eu ce besoin de le faire. C'est ce besoin-là qui représente leur motivation.

[N. Entwistle, 2003] définit la motivation comme une des caractéristiques influençant la manière d'apprendre des étudiants. Ceci dit, si elle influence la manière d'apprendre des étudiants, elle pourrait donc être un facteur important dans la réussite [M. Bruinsma, 2004] ou dans l'échec [R. J. Vallerand et R. Bissonnette, 1992].

Deux aspects de la motivation ont été traités spécifiquement : L'autodétermination et l'engagement.

##### V.4.1. L'autodétermination

Dans leur article, [Deci et Ryan, 2002] mentionnent trois types de motivation dans ce qu'ils ont appelé la « théorie de l'autodétermination » ou le « seuil de l'autodétermination » :

- La **motivation intrinsèque** qui exprime un choix fait avec plaisir, avec intérêt, de manière spontanée ;
- La **motivation extrinsèque** qui exprime un choix fait non en raison du plaisir qu'il procure mais pour des raisons externes
- **L'amotivation** qui est l'absence de la motivation.

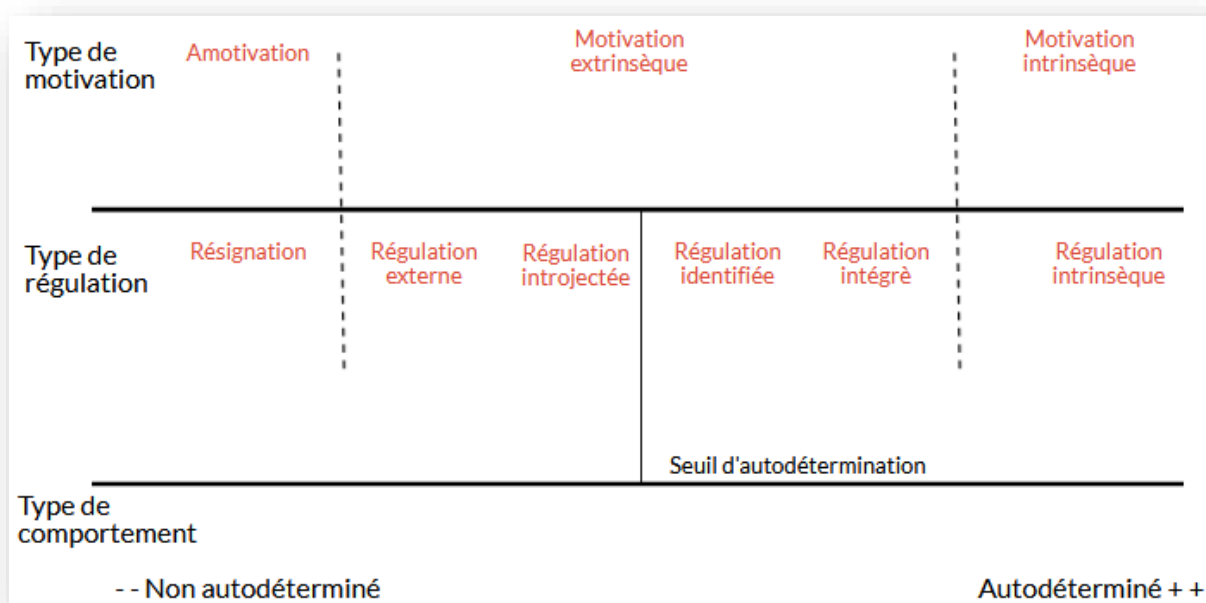
Une motivation est dite autodéterminée si elle traduit la capacité de l'acteur à se sentir responsable de ses choix plutôt que ceux-ci soient déterminés par des contraintes internes ou externes [Deci et Ryan, 1985]. Dans le seuil de l'autodétermination, l'amotivation est le niveau le plus bas et la motivation intrinsèque est le niveau le plus haut. Dans le cas de la motivation extrinsèque, le degré d'autodétermination est mitigé et est représenté par quatre types :

- **La motivation extrinsèque par régulation intégrée (MERI)** qui exprime un choix correspondant aux valeurs et qui est cohérent aux objectifs et aux aspirations de soi.  
*Ex : Un étudiant fait bien ses devoirs afin de réussir ses études en visant la finalité qu'un jour, il trouvera l'emploi de ses rêves*
- **La motivation extrinsèque par régulation identifiée (MEID)** exprime un choix fait non seulement par intérêt pour l'activité mais aussi pour la finalité qu'apporterait cette activité.  
*Ex : Un étudiant veut maîtriser l'algorithmique parce qu'il a pour but de devenir programmeur*
- **La motivation extrinsèque par régulation introjectée (MEIN)** exprime un choix fait sous l'influence d'un certain sentiment de culpabilité, d'une pression.  
*Ex : Un étudiant choisit la filière informatique parce qu'on lui a dit que le domaine de l'IT<sup>3</sup> était prometteur et cela alors qu'il aurait préféré faire les sciences humaines*
- **La motivation extrinsèque par régulation externe (MERE)** exprime un choix fait sous influence de la peur d'une punition, ou par recherche d'une récompense.  
*Ex : Un étudiant qui fait ses devoirs de peur d'être puni par son professeur.*

---

<sup>3</sup> IT (Information Technology) : Technologie de l'information

Figure V.1. Seuil de l'autodétermination<sup>4</sup>



Il est dès lors possible de déterminer si une motivation exprimée est auto-déterminante ou non, si elle pourrait aider l'étudiant à réussir ou pas.

#### V.4.2. Engagement vers un but.

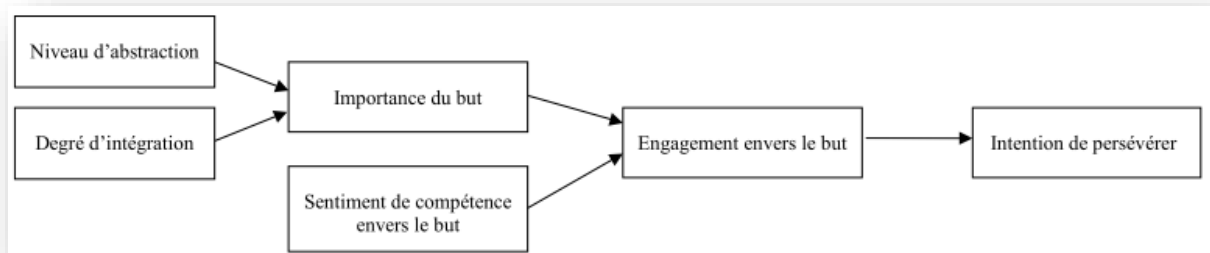
[Brunstein, 1993] et [Hollenbeck et Klein, 1987] ont défini *l'engagement envers un but* comme « le degré avec lequel un but particulier est associé à un fort sentiment de détermination, ainsi qu'à la volonté de faire des efforts pour l'atteindre ». D'autres part, [Germeijs & Verschueren, 2007] ont étudié l'engagement envers un but dans le contexte du processus de choix de filière chez les étudiants en dernière année du secondaire et leurs résultats montrent que, non seulement l'engagement envers un but est un prédicteur important de la satisfaction par rapport au choix de filière, mais il est aussi prédicteur de la performance et de l'adaptation dans les études choisies par l'étudiant. Cela équivaut à dire que plus le degré d'engagement d'un étudiant envers son objectif est faible, plus ça risque de compromettre sa réussite.

Cet engagement envers le but est lui-même influencé par l'importance du but aux yeux de l'étudiant et le sentiment de compétence selon le modèle proposé par [Boudrenghien et al,

<sup>4</sup> Source : <http://rire.ctreq.qc.ca/2019/01/les-differents-types-de-motivation-selon-la-theorie-de-lautodetermination/>

2013]. Dans ce modèle, il est fait mention de l'article de [Carver et Scheier, 1998], qui soutient que « l'importance d'un but est influencée par sa position au sein de la hiérarchie des buts d'une personne ». Cette position est définie par le **niveau d'abstraction du but et son degré d'intégration**.

Figure V.2. Engagement envers le but [Boudrenghien et al, 2013]



Carver et Scheier expliquent qu'un « but formulé à un *haut niveau d'abstraction* concerne le fait d'être ou de devenir un certain type de personne et s'applique généralement à *long terme* et un but formulé à un *faible niveau d'abstraction* concerne le fait de réaliser un certain type d'action et s'applique généralement à court terme » [Carver et Scheier, 1998]. Le degré d'intégration, pour sa part, est la mesure avec laquelle l'atteinte d'un but contribue à la poursuite d'autres buts au même niveau de la hiérarchie et à des niveaux plus élevés [Carver et Scheier, 1998].

En résumé, pour déterminer si un étudiant a l'intention de persévérer, le premier pas à faire serait d'évaluer le niveau d'abstraction de l'expression de son but ainsi que le degré d'intégration. Dans leur recherche, les auteurs [Boudrenghien et al, 2013] ont montré que plus le niveau d'abstraction du but est élevé, plus l'engagement envers ce but serait élevé, et ce, notamment sous l'effet d'une augmentation de l'importance du but. De même avec le degré d'intégration, s'il est élevé, l'engagement envers le but serait aussi élevé.

### Mesure du niveau d'abstraction

Selon les auteurs [Boudrenghien et al, 2013], l'expression d'un but peut être considéré comme de niveau d'abstraction élevé, faible ou moyen. Chacun des niveaux s'est vu attribuer un code respectivement (2, 0 ou 1) selon l'aspect dans lequel l'évaluation est faite. Deux aspects ont été considérés :

- L'aspect « be goal » ou « do goal »

- a. Si la formulation est liée au fait de devenir un certain type de personne « be goal » → **abstraction élevée et code = 2**
- b. Si la formulation est liée au fait de réaliser un certain type d'action « do goal » → **abstraction faible et code = 0**
- c. Si la formulation est liée au fait de réaliser un certain type d'action et au fait de devenir un certain type de personne → **abstraction moyenne et code = 1**
- L'aspect « long terme » / « court terme »
  - a. Si la formulation s'applique à long terme → **abstraction élevée et code = 2**
  - b. Si la formulation s'applique à court terme → **abstraction faible et code = 0**
  - c. Si la formulation comporte les deux niveaux d'abstraction → **abstraction moyenne et code = 1**

En mélangeant les deux aspects, un total sur 4 est obtenu pour une formulation « be goal » qui s'applique à « long terme ». Un exemple a été proposé dans leur travail pour illustrer ce système de codage.

*Tableau V.2. Exemples de buts de formation et du codage de leur niveau d'abstraction.  
[Boudrenghien et al, 2013]*

Faible niveau d'abstraction (code = 0)	Niveau moyen d'abstraction (code = 2)	Niveau élevé d'abstraction (code = 4)
1. Étudier à la « Louvain School of Management ».	1. Travailler dans le domaine de l'éducation.	1. Être reconnu dans mon travail.
2. Suivre des cours de math.	2. Soigner les gens qui en ont besoin.	2. Être ingénieur civil.
3. Étudier les différents médicaments.	3. Trouver un travail que j'aime.	3. Devenir quelqu'un d'important.
4. Obtenir un diplôme en psychologie.	4. Être passionné par mes études.	4. M'épanouir dans mon travail.
5. Réussir mes études.	5. Bien gagner ma vie.	5. Devenir une personne ouverte d'esprit.

#### V.4.3. Quelles sont ces motivations

La raison la plus évoquée comme source de motivation du choix des étudiants de la filière informatique est **l'intérêt** selon [Päivi et al, 2016] [Küllü Kori et al, 2016]. C'est un facteur intrinsèque. Il y a aussi les **opportunités** que peuvent offrir le secteur informatique [Küllü Kori et al, 2014]. D'autres raisons évoquées sont le **salaire et la sécurité** de l'emploi [Amnah Alshahrani, Isla Ross, and Murray Wood, 2018], *l'expérience dans la programmation*

[Küllü Kori et al, 2016] et dans l'informatique en général [Säde et al, 2019], entre autres. Dans leur étude, Külli et Al, ont pu regrouper 14 catégories des raisons du choix de la filière informatique, classées dans l'ordre :

Tableau V.3. Catégories des raisons du choix de la filière informatique [Säde et al, 2019]

<i>Catégories</i>	<i>Proportion</i>
<i>Intérêt</i>	55,4%
<i>Expérience précédent</i>	17,8%
<i>Développement personnel</i>	16,8%
<i>Réalisation de soi</i>	16,8%
<i>Importance dans le futur</i>	13,1%
<i>Marché du travail</i>	11,8
<i>Secteur en évolution</i>	9,4%
<i>Aime l'ICT</i>	8,5%
<i>Autre</i>	7,3%
<i>Continuer les études</i>	6%
<i>Convenance</i>	4,4%
<i>Salaire</i>	3,3%
<i>Nécessaire pour le travail</i>	2,8%
<i>Bourse d'études</i>	0,22



## VI. Développement de la recherche

### VI.1. La Recherche

#### VI.1.1. Rappel du contexte

Il a été constaté que les étudiants qui s'inscrivent en première année en INGMI ou en INFO à l'université n'avaient pas tous au départ les mêmes compétences en programmation. Une enquête leur a été soumise afin de prendre connaissance de leurs compétences. L'objectif derrière cette enquête est d'évaluer si ces compétences sont des facteurs qui influenceraient la réussite ou non au cours d'introduction à la programmation (INFOB131).

#### VI.1.2. Question de recherche

Dans le cadre de ce mémoire, la question de recherche posée est la suivante : *Quels sont les facteurs favorisant la réussite ou l'échec des étudiants de première année d'INGMI et d'INFO au cours d'introduction à la programmation de l'université de Namur ?*

Le but donc de ce travail consiste à identifier les différents profils existants parmi les étudiants d'INGMI et d'INFO sur base des réponses récoltées et de leurs notes obtenues à l'examen de première session janvier au cours d'introduction à la programmation.

Les profils des étudiants seront définis selon quatre groupes de compétence :

- *Bagage en informatique* : Ce sont les compétences en informatique acquises à l'école secondaire par l'étudiant
- *Bagage en programmation* : Ce sont les compétences en programmation de l'étudiant.
- *Bagage en Python* : Il s'agit de savoir si l'étudiant a déjà des connaissances à propos du langage de programmation « Python ».
- *Motivation* : Il s'agit du niveau de l'autodétermination et d'abstraction de l'étudiant.

Il s'agit premièrement d'analyser les compétences (issues des quatre groupes) qu'ont les étudiants et deuxièmement, de déterminer lesquelles de ces compétences favorisent ou pas la réussite au cours d'introduction à la programmation.

#### VI.1.3. Hypothèses

Pour apporter des éléments de réponse à la question de recherche, des hypothèses sont formulées, fortement liées aux compétences/capacités de chaque groupe défini.

Avant d'énoncer les hypothèses, il est important de rappeler l'influence de la pensée computationnelle par rapport à la réussite dans un cours d'introduction à la programmation. Les hypothèses en sont inspirées (cf. le point [V.3.2](#)). Nous avons, en effet, listé les cours informatiques qui sont enseignés en Belgique francophone ainsi que les compétences que sont

censés acquérir les étudiants. Et nous supposons que parmi ces cours, il y en a qui ont des compétences pouvant comprendre la pensée computationnelle.

### **Hypothèse 1, liée au bagage informatique :**

*« Les étudiants ayant eu des cours dont les compétences sont reconnues comme développant la pensée computationnelle en secondaire, sont plus susceptibles de réussir le cours d'introduction à la programmation que ceux n'en ayant pas eu. »*

### **Hypothèse 2, liée au bagage en programmation :**

*« Les étudiants ayant suivi des cours d'algorithme et de programmation en secondaire sont plus susceptibles de réussir le cours d'introduction à la programmation que les autres n'en ayant pas eu. »*

### **Hypothèse 3, liée au bagage en Python :**

*« Les étudiants ayant déjà rencontré le langage Python ont plus de chance de réussir le cours de Programmation que les autres ne l'ayant pas rencontré. »*

### **Hypothèse 4, liée à la motivation :**

*« Les étudiants qui n'ont pas suffisamment de motivation vis-à-vis de l'apprentissage de la programmation sont moins susceptibles de réussir le cours d'introduction à la programmation que ceux qui ont une motivation à apprendre à programmer. »*

## **VI.2. Les données**

### **VI.2.1. Enquête et Population**

L'enquête utilisée dans ce mémoire a été développée en 2015, dans le cadre d'une thèse de Doctorat, et est soumise aux étudiants-entrant d'INGMI et d'INFO de l'université de Namur depuis l'année académique 2015-2016. Cependant les données collectées ne concernent que celles de la version du questionnaire utilisé durant les années académiques 2017-2018, 2018-2019, 2019-2020 et 2020-2021 qui est plus complet que le questionnaire des années académiques 2015-2016 et 2016-2017.

Cette enquête a pour but de mettre en évidence, pour chaque étudiant, l'existence (ou l'absence) d'un bagage en programmation en début d'année académique et d'essayer de comprendre pourquoi le cours de programmation pose autant de difficultés à certains étudiants.

La participation à ce questionnaire est obligatoire et l'objectif final est d'aider au mieux les étudiants éprouvant une certaine difficulté dans le cours.

En annexe se trouve le questionnaire de 20 questions passé tel quel auprès des étudiants. Il compte des questions ouvertes, des questions à option de réponse, etc.

Les étudiants de première année d'INGMI et d'INFO de l'université de Namur ont répondu au questionnaire au moment de leur inscription ou réinscription en première année durant les années académiques 2017-2018, 2018-2019, 2019-2020 et 2020-2021.

Parmi ces étudiants, il existe différents statuts : les « nouveaux » directement issus de l'enseignement secondaire, les « bisseurs » qui reprennent l'année dans la même faculté, ceux qui changent d'orientation et viennent donc d'autres facultés de l'université de Namur (autres filières) et enfin il y a ceux qui viennent qui décident de choisir la filière INGMI ou INFO et enfin, il y a ceux qui sont en situation d'étalement ou proviennent d'autres universités (autres cas).

Il est important de signaler que les noms des étudiants ayant répondu à cette enquête n'ont jamais été communiqués au Professeur du cours d'introduction à la programmation. Les données qui ont été collectées lors de cette étude et les résultats qui en ont découlés, ont été rendus anonymes.

#### VI.2.2. La base de données

Les données obtenues suite à la soumission du questionnaire aux étudiants, forment la base des données de ce mémoire. Ci – dessous, ces données sont présentées en tant que variables, catégorisées dans différents groupes, les mêmes que ceux mentionnés dans les hypothèses de recherche : groupe « *bagage informatique* », groupe « *bagage en programmation* », groupe « *bagage en python* » et groupe « *motivation* ».

Pour chacun de ces groupes de variables, un tableau de quatre colonnes est réalisé, spécifiant le numéro de la question (voir Annexe) pour laquelle est créée la variable, le nom de la variable, le type de variable et les réponses possibles (variables nominales). Selon [Ganassali, 2014)], il existe quatre<sup>5</sup> types de variable :

- **Les variables numériques** : Ce sont des nombres provenant des mesures précises. On peut les additionner, les soustraire, les diviser et les multiplier en obtenant un résultat de même nature.

---

<sup>5</sup> Les descriptions de ces variables sont tirées du livre de Jean MOSCAROLA « *Faire parler les données. Méthodologies quantitatives et qualitatives*. EMS Editions, 2018 »

- **Les variables échelles ou ordinales** : elles correspondent à des ordres de grandeur discontinus tels que tranche d'âge, niveau de satisfaction, etc. Ces variables sont utilisées lorsqu'une mesure précise est inapplicable. Selon le besoin, elles peuvent être considérées comme des variables numériques ou nominales.
- **Les variables nominales** : Ce sont des variables dites qualitatives ou catégorielles telles que les genres (féminin, masculin) ou des catégories socio-professionnelles (ouvrier, employé, cadre). Chaque état de la variable nominale est décrit par un mot ou un code qui en définit les modalités. La seule opération possible sur ce type de variable est le calcul des effectifs de chaque modalité.
- **Les variables textes** : De nature différente que les autres variables car il faut les lire, ces variables sont transformables en variables nominales ou numériques mais avec le risque d'en perdre le sens. Pour traiter ces données, il existe donc des méthodes spécifiques d'analyse sémantique, textuelle.

Les réponses possibles des questions 2, 3, 13 et 14 ont été modifiées pour les besoins de simplification et d'analyse.

Les modifications apportées aux réponses à la question 2 sont les suivantes :

- La réponse INGMI remplace les réponses « INGMI 1-3 » et « INGMI 2-4 »
- La réponse INFO remplace les réponses « INFOA » et « INFOB »

Les modifications apportées aux réponses à la question 3 sont les suivantes :

- La réponse « Autres cas » remplace les réponses « Non, je suis en situation d'étalement » et « Autre »
- La réponse INFO remplace les réponses « INFOA » et « INFOB »

Les modifications apportées aux réponses à la question 13 sont les suivantes :

- La réponse « Oui » remplace la réponse « J'ai eu cours d'info »
- La réponse « Non » remplace les réponses « Je n'ai pas eu cours d'info » et « Je ne m'en souviens pas »

Les modifications apportées aux réponses à la question 14 sont les suivantes :

- La réponse « Oui » remplace les réponses « Je suis certaine que Oui » et « Je pense que Oui »
- La réponse « Non » remplace les réponses « Je suis certain que Non », « Je pense que non » et « Je ne me souviens pas »

Tableau VI.1. Variables liées au Bagage Informatique de l'étudiant

Questions	Variable	Type	Réponses possibles
Question 11	cours_info_sec	Nominal	Oui / Non
Question 13	info[1-6]sec	Nominal	Oui / Non
Question 14	cours_codage_info	Nominal	Oui / Non
Question 14	cours_ordi_arch	Nominal	Oui / Non
Question 14	cours_os	Nominal	Oui / Non
Question 14	cours_reseau	Nominal	Oui / Non
Question 14	cours_html	Nominal	Oui / Non
Question 14	cours_db	Nominal	Oui / Non

Tableau VI.2. Variables liées au Bagage en Programmation de l'étudiant

Questions	Variable	Type	Réponses possibles
Question 4	lang_list	Texte	-
Question 15	cours_others	Texte	-
Question 17	autod_prog_list	Texte	-
Question 14	cours_algo	Nominal	Oui / Non
Question 14	cours_programmation	Nominal	Oui / Non

Tableau VI.3. Variables liées au Bagage en Python de l'étudiant

Questions	Variable	Type	Réponses possibles
Question 4	lang_list	Texte	-
Question 15	cours_others	Texte	-
Question 17	autod_prog_list	Texte	-

Tableau VI.4. Variables liées à la motivation de l'étudiant

Questions	Variable	Type	Réponses possibles
Question 5	raison_filière	Texte	-
Question 16	autodidacte_prog	Nominal	Oui / Non
Question 18	cours_prepa	Nominal	Oui / Non

Les variables citées ci – dessous présentent l'étudiant : son sexe, la faculté dans laquelle il est inscrit et son statut, c'est-à-dire nouvellement inscrit, biceur, etc. etc.

Tableau VI.5. Autres Variables liées à l'étudiant

Questions	Variable	Type	Réponses possibles
-	sexe	Nominal	M / F
Question 2	fac	Nominal	INGMI / INFO
Question 3	statut	Nominal	Oui / Non, je bice ma première année / Non, j'ai effectué une première année dans une autre filière / Autre cas

La dernière des variables à présenter est la note que l'étudiant a obtenu à son examen lors de la session de janvier (première session). Cette donnée ne fait pas partie des données récoltées via l'enquête.

Tableau VI.6. Variable représentant la note de l'étudiant à l'examen de Programmation

Question	Variable	Type	Réponses possibles
-	<i>note_janvier</i>	Numérique	-

### VI.2.3. Traitement des données

#### 1. La note de l'examen

Les notes obtenues par les étudiants à l'examen d'introduction à la programmation (cf. [Tableau VI.6](#)) et la variable « *note\_janvier* » qui en résulte, sont de type « numérique ». Pour des besoins d'analyse, cette variable doit être de type « catégorielle ». Une variable « note » a donc été créée, pouvant prendre la valeur 0 ou 1. La valeur 0 (échec) est donc attribuée à des notes d'examen inférieures à 10 et la valeur 1 (réussite) aux notes supérieures ou égales à 10.

Tableau VI.7. Nouvelle variable dépendante nominale

Variable	Valeurs possibles	Description
<i>note</i>	0 ou 1	0 → échec c'est-à-dire $note\_janvier < 10$ 1 → réussite c'est-à-dire $note\_janvier \geq 10$

#### 2. Variables – Bagage informatique

Des changements ont aussi été apportés aux variables liées au bagage informatique de l'étudiant (cf. Tableau VI.1). Les variables *info1sec*, *info2sec*, ... *info6sec* ont été remplacées par une nouvelle variable nominale « ***bg\_info\_56*** » qui prend la valeur « **Oui** » si l'étudiant a eu cours d'informatique en cinquième **ou** en sixième secondaire et « **Non** » dans les autres cas. La cinquième et la sixième secondaire représentant le *troisième degré* dans le programme informatique (cf. le document source du point [V.3.2](#)), à ce niveau, les compétences informatiques sont des vraies compétences.

Tableau VI.8. Nouvelles variables pour le bagage informatique

Variable	Valeurs possibles	Description
<i>cours_info_sec</i>	Oui ou Non	Oui → informatique au secondaire Non → Pas d'informatique au secondaire
<i>bg_info_56</i>	Oui ou Non	Oui → informatique en 5 <sup>e</sup> ou 6 <sup>e</sup> secondaire Non → Pas d'informatique en 5 <sup>e</sup> et 6 <sup>e</sup>
<i>cours_codage_info</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi
<i>cours_ordi_arch</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi
<i>cours_os</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi
<i>cours_reseau</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi

<i>cours_html</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi
<i>cours_db</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi

### 3. Variables – Bagage en Programmation et en Python

Les variables *lang\_list*, *cours\_others* et *autod\_prog\_list* (cf. Tableau VI.2), dans lesquelles les étudiants ont eu à citer les langages de programmation qu'ils auraient appris avant le début de l'année académique, ont été remplacées par une seule variable « **langage** » qui prend la valeur 1 si l'étudiant a déjà appris au moins un langage de programmation et 0 dans d'autres cas.

Tableau VI.9. Nouvelles variables pour le Bagage en programmation

Variable	Valeurs possibles	Description
<i>cours_algo</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi
<i>cours_programmation</i>	Oui ou Non	Oui → cours suivi Non → cours non suivi
<i>langage</i>	0 ou 1	0 → 0 langage de programmation 1 → ≥ 1 langage de programmation

Ce même changement a été apporté aux trois variables liées au bagage en Python (cf. [Tableau VI.3](#)). Ces trois variables ont été remplacées par la variable « **python** » qui prend la valeur 1 si l'étudiant a déjà rencontré au moins une fois le langage Python, sinon la valeur 0.

Tableau VI.10. Nouvelle variable pour le bagage en Python

Variable	Valeurs possibles	Description
<i>python</i>	0 ou 1	0 → Pas de connaissance du langage Python 1 → Connaissance du langage Python

### 4. Variables - Motivation

Un travail conséquent a été réalisé sur les variables liées à la motivation (cf. Tableau VI.4). Pour rappel, il avait été demandé aux étudiants de donner la raison de leur choix de filière (informatique ou ingénierie de gestion en management de l'information). Dans un premier temps, les raisons citées ont été catégorisées. Ensuite, deux variables ont été ajoutées, liées à l'autodétermination et au niveau d'abstraction.

#### Catégorisation des raisons

Une analyse lexicale « Text mining », au moyen de R, des réponses obtenues à la question ouverte a été réalisée.





Une deuxième étape a consisté à identifier les associations de terme. Dans la Figure VI.3, il apparaît que le mot « *mond* » a été cité 44 fois parmi lesquelles 36 fois au côté du mot « *informatiqu* ». Le mot « *interest* » a été cité 59 fois sur 125 aux côtés du mot « *informatiqu* ».

Figure VI.3. Résultat de Co – occurrence du terme "informatiqu"

```
> cooc_terms(dtms0,"informatiqu")
```

	% Term/Cooc.	% Cooc./Term	Global %	Cooc.	Global	t value	Prob.
informatiqu	10.10464356	100.00000	5.98025934	309	309	Inf	0.0000
mond	1.17724003	81.81818	0.85155796	36	44	3.042510	0.0012
pens	0.78482668	80.00000	0.58060770	24	30	2.210220	0.0135
plus	3.98953564	66.66667	3.54170699	122	183	2.037243	0.0208
base	0.42511445	86.66667	0.29030385	13	15	1.989725	0.0233
outil	0.22890778	100.00000	0.13547513	7	7	1.953816	0.0254
choisi	0.71942446	78.57143	0.54190052	22	28	1.950777	0.0255
connaiss	0.32701112	90.90909	0.21288949	10	11	1.932121	0.0267
indispens	0.19620667	100.00000	0.11612154	6	6	1.718126	0.0429
-----	NA	NA	NA	NA	NA	NA	NA
convaincu	0.00000000	0.00000	0.05806077	0	3	-1.491283	0.0679
technologi	0.26160889	40.00000	0.38707180	8	20	-1.511246	0.0654
choix	0.09810334	30.00000	0.19353590	3	10	-1.547248	0.0609
découvert	0.06540222	25.00000	0.15482872	2	8	-1.598920	0.0549
info	0.06540222	25.00000	0.15482872	2	8	-1.598920	0.0549
résoudre	0.03270111	16.66667	0.11612154	1	6	-1.697937	0.0448
filièr	0.71942446	45.83333	0.92897232	22	48	-1.731433	0.0417
développ	0.13080445	30.76923	0.25159667	4	13	-1.791402	0.0366
ça	0.65402224	44.44444	0.87091155	20	45	-1.854173	0.0319
où	0.06540222	20.00000	0.19353590	2	10	-2.195734	0.0141
voulai	0.32701112	35.71429	0.54190052	10	28	-2.319764	0.0102
créer	0.16350556	27.77778	0.34836462	5	18	-2.459169	0.0070
interest	1.92936560	47.20000	2.41919876	59	125	-2.645436	0.0041

Enfin, avec la liste des mots les plus cités ainsi que les co-occurrences des différents termes, il est possible d'entrevoir les raisons évoquées par les étudiants. Celles-ci ont été catégorisées au mieux en regard à la question de recherche (cf. [Tableau VI.7](#)). Pour confirmer ce résultat, chacune des réponses collectées a été lue afin d'en saisir le vrai sens, d'être le plus précis possible et d'exprimer au mieux l'intention de l'étudiant.

Tableau VI.11. Catégories des raisons de choix de la filière informatique

Catégories	Caractéristiques
Apprendre	Parce que l'étudiant est curieux Veut apprendre Veut découvrir Aime chipoter Veut améliorer ses compétences
Avenir	Parce que l'étudiant se dit que l'IT est un : Métier d'avenir Domaine d'avenir Secteur d'avenir
Carrière	Parce que l'étudiant

	Veut un jour travailler dans le domaine IT. Trouve en l'IT une opportunité de carrière
<i>Défaut</i>	Parce que l'étudiant N'a pas eu d'autres choix Veut éviter d'autres facs N'a pas donné de réponse
<i>Facilité</i>	Parce que l'étudiant Connait déjà très bien le domaine de l'IT A eu des cours d'informatique en secondaire A pratiqué depuis longtemps Pense avoir un atout pour réussir Pense que c'est le domaine qui lui convient le mieux Reprend l'année S'y sent plus à l'aise
<i>Impact</i>	Parce que l'informatique Apporte beaucoup à la société Aide à résoudre des problèmes de la société Fait partie intégrante de la vie quotidienne Est incontournable dans le monde d'aujourd'hui
<i>Intérêt</i>	Parce que l'étudiant Veut faire partie de l'avancement de l'informatique Veut combiner avec autre domaine Aime l'informatique Aime les Maths ou la logique ou les jeux, ...
<i>Passion</i>	Parce que l'étudiant Dit que c'est une passion pour lui
<i>Programmation</i>	Parce que l'étudiant Est intéressé par la programmation

## Autodétermination

Une nouvelle variable exprimant le niveau d'autodétermination a été ajoutée dans la liste des variables concernant la motivation. Comme discuté au point sur l'autodétermination dans l'état de l'art (cf. [V.4.1](#)), il existe trois types de motivation : intrinsèque, extrinsèque et l'amotivation.

Les raisons du choix de la filière informatique données par les étudiants ont été parcourues et classées en trois groupes (cf. [Figure V.1](#)) : celles considérées comme des motivations intrinsèques c'est-à-dire d'autodétermination très élevée, celles qui reflètent un niveau moyen positif d'autodétermination (MERI et MEID) et celles qui n'expriment pas d'autodétermination (MEIN, MERE et l'amotivation). Cette nouvelle variable prendra la valeur 0 pour une absence d'autodétermination, la valeur de 1 pour une autodétermination moyenne et 2 pour une autodétermination élevée. Le tableau VI.12 présente des réponses données par des étudiants et leur catégorisation dans un seuil d'autodétermination.

*Tableau VI.12. Exemples des raisons données par seuil d'autodétermination*

*Autodétermination Exemples*

<i>Élevée (code = 2)</i>	« Je suis passionné par l'informatique et ça correspond mieux à ce que je fais dans mon quotidien, et je souhaite m'améliorer »  « C'est un domaine que j'aime et qui m'intéresse »
<i>Moyenne (code = 1)</i>	« L'informatique est un bagage à savoir utiliser et maîtriser pour entrer dans le monde du travail » (MEID)  « J'aimerais pouvoir développer des jeux vidéos ou des programmes » (MERI)
<i>Inexistante (code = 0)</i>	« Car je ne voulais pas de science dure » (MEIN)  « Je suis en liste d'attente d'infographie dans une autre école et l'informatique était mon second choix » (MEIN)  « 0 » (Amotivation)

### Niveau d'abstraction

En ce qui concerne le niveau d'abstraction, une nouvelle variable « **abstraction** » a aussi été créée afin d'exprimer d'une autre manière la motivation conformément à ce qui a été dit dans l'état de l'art (cf. [V.4.2](#)). Le même système de codage que celui montré par le Tableau V.2 pour chaque raison donnée par l'étudiant a été adopté. La valeur de 4 a été attribuée pour un niveau d'abstraction élevé (*Be goal/Long terme*), la valeur de 2 (« *Be goal* » / « *Court Terme* » ou « *Do Goal* » / « *Long terme* ») pour un niveau d'abstraction moyen et 0 (« *Do goal* » / « *Court Terme* ») pour un niveau d'abstraction moyenne.

Comme pour l'autodétermination, le tableau VI.13 présente des raisons données par les étudiants et les codes associées liés aux niveaux d'abstractions.

Tableau VI.13. Exemples de codage du niveau d'abstraction

Niveau d'abstraction	Exemples
<i>Élevé (code = 4)</i>	« Parce que j'aimerais développer pour Android et que les possibilités sont infinies. De plus c'est un métier d'avenir. »  « Cela concorde avec mon projet professionnel qui est de travailler dans le domaine de la cybersécurité. »
<i>Moyen (code = 2)</i>	« J'ai besoin de cette filière pour mon futur travail. »  « J'ai fait ce choix car je le trouvais intéressant pour mon futur »
<i>Faible (code = 0)</i>	« J'aime ce qu'il se passe dans l'ordinateur et autour de lui »  « J'avais envie de faire de l'informatique »

## Résumé des variables liées à la motivation

Finalement, cinq variables sont à considérer pour la motivation.

Tableau VI.14. Nouvelle liste des variables pour la motivation

Variable	Valeurs possibles	Description
<i>raison_filière</i>	Au moins une des catégories du Tableau VI.11	Une raison peut appartenir à plus d'une catégorie.
<i>autodidacte_prog</i>	Oui ou Non	Oui → Apprentissage autodidacte Non → Pas d'apprentissage autodidacte
<i>cours_prepa</i>	Oui ou Non	Oui → participation aux cours préparatoires Non → pas de participation aux cours préparatoires
<i>autodétermination</i>	0, 1 ou 2	0 → autodétermination élevée 1 → autodétermination moyenne 2 → autodétermination inexistante
<i>abstraction</i>	0, 2 ou 4	0 → niveau d'abstraction faible 2 → niveau d'abstraction moyen 4 → niveau d'abstraction élevé

### VI.2.4. Répartition des étudiants

#### 1. Répartition globale

Les étudiants ayant répondu à l'enquête et dont les réponses ont été jugées complètes et utilisables dans le cadre de l'étude, sont au nombre 489 : 201 étudiants en INGMI et 288 étudiants en INFO. La répartition est la suivante :

Tableau VI.15. Répartition globale des étudiants

Répartition par faculté		Répartition par sexe	
INGMI	INFO	Féminin	Masculin
201 (41.1%)	288 (58.9%)	92 (18.8%)	397 (81.2%)
Répartition par statut			
Nouveau	Bisseur	Autre filière	Autre cas
421 (86.1%)	13 (2.6%)	40 (8.2%)	15 (3.1%)

Tableau VI.16. Répartition des étudiants d'INGMI

Répartition par Sexe			
Féminin		Masculin	
64 (31.8%)		137 (68.2%)	
Répartition par statut			
Nouveau	Bisseur	Autre filière	Autre cas
176 (87.5%)	5 (2.5%)	15 (7.5%)	5 (2.5%)

Tableau VI.17. Répartition des étudiants d'INFO

Répartition par Sexe	
Féminin	Masculin

28 (9.7%)		260 (90.3%)	
Répartition par statut			
Nouveau	Bisseur	Autre filière	Autre cas
245 (85%)	8 (2.8%)	25 (8.7%)	10 (3.5%)

Le taux de réussite pour la population INGMI est de 68% (137/201), contre 67% (192/288) pour la population INFO.

La population INGMI est constituée de 201 étudiants. 75% des filles (48/64) ont réussi le cours d'introduction à la programmation. Chez les garçons, le taux de réussite est de 65% (89/137). Au niveau du statut, le taux de réussite est de 67% chez les « *nouveaux* » étudiants. Un *bisseur* sur cinq a échoué, comme un tiers des étudiants (5 sur 15) provenant d'autres filières. Enfin, les cinq étudiants faisant partie des « *autres cas* » ont tous réussi au cours d'introduction à la programmation.

La population INFO est composée de 288 étudiants. Près de la moitié des filles (13/28) ont raté leur examen de janvier du cours d'introduction à la programmation (taux de réussite 53.6%), comme 68.5% des garçons (82/260). Au niveau du statut, 32.7% d'étudiants « *nouveaux* » (80/245) étaient en échec à l'examen de janvier en programmation. C'est 50/50 pour les huit bisseurs. Quant aux étudiants provenant d'une autre filière, 16 (sur 25, soit 64.0%) ont pu valider la matière pour l'année. Enfin, les étudiants faisant partie des « *autres cas* » de statut ont un taux de réussite de 80% (8/10).

## 2. Répartition au niveau des variables du bagage informatique

Il s'agit ici d'établir la proportion d'étudiants ayant un bagage informatique selon les variables qui ont été établies précédemment au [Tableau VI.8](#).

Concernant le bagage informatique, la variable qui définit le nombre d'étudiants concernés par ce bagage est « *cours\_info\_sec* ». Sur les 489 étudiants : 112 étudiants, soit 22.9%, ont répondu avoir eu des cours d'informatique au secondaire « *cours\_info\_sec* = Oui ».

*Tableau VI.18. Répartition des étudiants ayant eu des cours d'informatique en secondaire*

Répartition par faculté		Répartition par sexe	
INGMI	INFO	Féminin	Masculin
20	92	17	95
Répartition par statut			
Nouveau	Bisseur	Autre filière	Autre cas
102	2	6	2

Il est constaté que pour la fac INGMI, seulement 10% (20 sur 201 étudiants, dont 6 filles et 14 garçons) des étudiants ont eu au moins un cours d'informatique à l'école secondaire contre 32% (92 sur 288 étudiants, dont 11 filles et 81 garçons) pour les étudiants en INFO.

La répartition du nombre d'étudiants par variable du bagage informatique est la suivante :

Tableau VI.19. Répartition du nombre d'étudiants par variable du bagage informatique

Variable	INGMI			INFO		
	Filles	Garçons	Total	Filles	Garçons	Total
<i>cours_info_sec</i>	6	14	20	11	81	92
<i>bg_info_56</i>	5	7	12	6	67	73
<i>cours_codage_info</i>	1	2	3	6	54	60
<i>cours_ordi_arch</i>	2	9	11	6	64	70
<i>cours_os</i>	0	7	7	5	56	61
<i>cours_reseau</i>	3	9	12	6	41	47
<i>cours_html</i>	3	2	5	7	65	72
<i>cours_db</i>	3	3	6	2	47	49

Les tableaux ci-dessous (cf. Tableau VI.20 et Tableau VI.21) représentent, pour chaque colonne « Oui » (cours appris) ou « non » (cours pas appris), le nombre d'étudiants ayant réussi et celui des étudiants ayant échoué.

Pour la population INGMI, le Tableau VI.20 montre qu'il y a une répartition 50/50 de réussite et d'échec au cours d'introduction à la programmation pour la variable « *bg\_info\_56* » pour les étudiants ayant eu des cours d'informatique au troisième degré du secondaire. Six échecs et cinq réussites pour les étudiants ayant eu le cours d'architecture de base de l'ordinateur « *cours\_ordi\_arch* ». Le tableau montre encore qu'il y a eu plus d'échecs (4) que de réussite (2) pour les étudiants ayant eu de cours de base de données « *cours\_db* » en secondaire, et un taux de réussite de réussite de 80% pour les étudiants ayant un cours de création de pages Web « *cours\_html* ».

Tableau VI.20. Nombre de réussite et d'échec pour le Bagage Informatique (INGMI)

Réponses	Oui		Non	
	Réussite	Échec	Réussite	Échec
<i>cours_info_sec</i>	11	9	126	55
<i>bg_info_56</i>	6	6	131	55
<i>cours_codage_info</i>	2	1	135	63
<i>cours_ordi_arch</i>	5	6	132	58
<i>cours_os</i>	4	3	133	61
<i>cours_reseau</i>	8	4	129	60
<i>cours_html</i>	4	1	133	63
<i>cours_db</i>	2	4	135	60

Pour la population INFO, le Tableau VI.21 montre des taux de réussite au cours d'introduction à la programmation de plus de 70% pour des étudiants ayant suivi les cours de codage d'information « *cours\_codage\_info* » (75%), de réseau et Internet « *cours\_reseau* » (70%), de création de pages Web « *cours\_html* » (74%) et de bases de données « *cours\_db* » (75,5%). 54 étudiants des 73 (74%) ayant eu des cours d'informatique au troisième degré du secondaire « *bg\_info\_56* » ont validé le cours d'introduction à la programmation.

Tableau VI.21. Nombre de réussite et d'échec pour le Bagage Informatique (INFO)

Réponses	Oui		Non	
	Réussite	Échec	Réussite	Échec
<i>cours_info_sec</i>	63	29	130	66
<i>bg_info_56</i>	54	19	139	76
<i>cours_codage_info</i>	45	15	148	80
<i>cours_ordi_arch</i>	48	22	145	73
<i>cours_os</i>	41	20	152	75
<i>cours_reseau</i>	33	14	160	81
<i>cours_html</i>	53	19	140	75
<i>cours_db</i>	37	12	156	83

Les taux de réussite pour les étudiants d'INGMI et d'INFO n'ayant pas suivi chacun des cours informatiques, sont tous supérieurs à 65%.

### 3. Répartition au niveau des variables du bagage en programmation

Il y a trois variables pour le bagage en programmation (Voir [Tableau VI.9](#)). Concernant le cours d'algorithme et le cours de programmation, il s'agit de savoir combien, sur les 112 étudiants ayant eu des cours d'informatique au secondaire, ont eu chacun de ces deux cours. 190 étudiants sur les 489 annoncent avoir déjà appris un langage de programmation « *langage* ».

Tableau VI.22. Répartition du nombre d'étudiants par variable du bagage en programmation

Variable	INGMI			INFO		
	Filles	Garçons	Total	Filles	Garçons	Total
<i>cours_algo</i>	2	1	3	4	51	55
<i>Cours_programmation</i>	5	3	8	4	65	69
<i>langage</i>	9	21	30	11	149	160

Sur les 112, il y a donc 68.7% (77/112) qui ont eu un cours de programmation et 51.8% (58/112) ont eu un cours d'algorithme à l'école secondaire

Le Tableau VI.23 montre un taux de réussite de 73.3% au cours d'introduction à la programmation pour des étudiants d'INGMI ayant appris au moins un langage de



programmation « langage ». Les étudiants ayant eu cours de programmation en secondaire ont un taux de réussite à 50% et ceux ayant eu cours d’algorithme, un taux de réussite à 0%.

Tableau VI.23. Nombre de réussite et d’échec pour le Bagage Programmation (INGMI)

Réponses	Oui/1		Non/0	
	Réussite	Échec	Réussite	Échec
<i>cours_programmation</i>	4	4	133	60
<i>cours_algo</i>	0	3	137	61
<i>langage</i>	22	8	115	56

Pour la population d’INFO, le Tableau VI.24 montre des taux supérieurs à 76% pour les étudiants ayant eu des cours de programmation « *cours\_programmation* » (77%), d’algorithme « *cours\_algo* » (76%) et pour les étudiants ayant déjà appris au moins un langage de programmation « *langage* » (77.5%).

Tableau VI.24. Nombre de réussite et d’échec pour le Bagage Programmation (INFO)

Réponses	Oui/1		Non/0	
	Réussite	Échec	Réussite	Échec
<i>cours_programmation</i>	53	16	140	79
<i>cours_algo</i>	42	13	151	82
<i>langage</i>	124	36	69	59

Les taux de réussite des étudiants d’INGMI et d’INFO n’ayant pas suivi le cours de programmation sont en moyenne de 65%, c’est autant que pour le cours d’algorithme. Ce n’est pas le cas pour les étudiants n’ayant appris aucun langage de programmation : ceux d’INGMI ont un taux de réussite de 63% tandis que ceux d’INFO ont un taux de 54%.

#### 4. Répartition au niveau des variables du bagage en Python

Parmi les 190 étudiants qui annoncent avoir déjà appris un langage de programmation, 89 d’entre eux annoncent avoir déjà appris le langage de programmation « Python ».

Tableau VI.25. Répartition du nombre d’étudiants par variable du bagage en Python

Variable	INGMI			INFO		
	Filles	Garçons	Total	Filles	Garçons	Total
<i>python</i>	5	8	13	3	73	76

Le Tableau VI.26 et le Tableau VI.27 montrent un taux de réussite pour les étudiants d’INGMI ayant mentionné avoir appris le langage Python, de 69% et celui des étudiants



d'INFO est de 74%. Ceux qui n'en ont pas fait, ont un taux de réussite de 68% pour les étudiants d'INGMI et de 63.7% pour INFO.

Tableau VI.26. Nombre de réussite et d'échec pour le Bagage Python (INGMI)

Réponses	1 (Oui)		0 (Non)	
	Réussite	Échec	Réussite	Échec
python	9	4	128	60

Tableau VI.27. Nombre de réussite et d'échec pour le Bagage Python (INFO)

Réponses	1 (Oui)		0 (Non)	
	Réussite	Échec	Réussite	Échec
python	58	18	135	77

## 5. Répartition au niveau des variables de Motivation

Les raisons de choix de filière citées par les étudiants ont été groupées dans des catégories étaient groupées dans des catégories (voir [Tableau VI.11](#)). La répartition globale des deux facultés est la suivante :

Figure VI.4. Fréquence des raisons les plus citées par les étudiants

```
> frequent_terms(dtm,n=9)
      global occ.  global %
intérêt      237 31.061599
apprendre     96 12.581913
carrière      93 12.188729
programmation  90 11.795544
impact        70  9.174312
avenir        64  8.387942
passion       50  6.553080
facilité      40  5.242464
défaut        23  3.014417
```

Dans la figure ci-dessus, il apparaît que l'intérêt est la raison la plus citée par les étudiants et est présente dans 32% des réponses. Pour chaque catégorie, il est possible de voir la répartition entre étudiants d'INGMI et d'INFO :

Figure VI.5. Fréquence par faculté des raisons les plus citées

```
> frequent_terms(dtm,meta(corpus)$fac,n=9)
```

\$INFO	% Term/Level	% Level/Term	Global %	Level	Global	occ.	t value	Prob.
intérêt	32.692308	64.55696	31.061599	153		237	1.1465257	0.1258
apprendre	13.034188	63.54167	12.581913	61		96	0.3566086	0.3607
programmation	12.820513	66.66667	11.795544	60		90	0.9902000	0.1610
carrière	10.897436	54.83871	12.188729	51		93	-1.2564611	0.1045
passion	9.829060	92.00000	6.553080	46		50	4.8855789	0.0000
facilité	7.905983	92.50000	5.242464	37		40	4.3835836	0.0000
avenir	6.410256	46.87500	8.387942	30		64	-2.3225398	0.0101
impact	4.914530	32.85714	9.174312	23		70	-4.9353406	0.0000
défaut	1.495726	30.43478	3.014417	7		23	-2.8331159	0.0023

\$INGMI	% Term/Level	% Level/Term	Global %	Level	Global	occ.	t value	Prob.
intérêt	28.474576	35.44304	31.061599	84		237	-1.1465257	0.1258
impact	15.932203	67.14286	9.174312	47		70	4.9353406	0.0000
carrière	14.237288	45.16129	12.188729	42		93	1.2564611	0.1045
apprendre	11.864407	36.45833	12.581913	35		96	-0.3566086	0.3607
avenir	11.525424	53.12500	8.387942	34		64	2.3225398	0.0101
programmation	10.169492	33.33333	11.795544	30		90	-0.9902000	0.1610
défaut	5.423729	69.56522	3.014417	16		23	2.8331159	0.0023
passion	1.355932	8.00000	6.553080	4		50	-4.8855789	0.0000
facilité	1.016949	7.50000	5.242464	3		40	-4.3835836	0.0000

Sur les 237 mentions de l'intérêt comme raison, les étudiants d'INFO l'ont mentionné 153 fois soit 64,5%. Les étudiants d'INFO évoquent également plus la « passion » (46 sur 50, soit 92%) et la « facilité » (37 sur 40, soit 92.5%) comme raisons de leur choix. La programmation est en troisième position en INFO et a moins la cote en INGMI (6<sup>ème</sup> position).

Figure VI.6. Fréquence par sexe des raisons les plus citées

```
> frequent_terms(dtm,meta(corpus)$sexe,n=9)
```

\$F	% Term/Level	% Level/Term	Global %	Level	Global	occ.	t value	Prob.
intérêt	30.2158273	17.72152	31.061599	42		237	-0.1288469	0.4487
carrière	15.8273381	23.65591	12.188729	22		93	1.2932207	0.0980
avenir	15.1079137	32.81250	8.387942	21		64	2.8184192	0.0024
impact	11.5107914	22.85714	9.174312	16		70	0.8990844	0.1843
apprendre	10.0719424	14.58333	12.581913	14		96	-0.8387528	0.2008
programmation	6.4748201	10.00000	11.795544	9		90	-2.0974495	0.0180
défaut	5.7553957	34.78261	3.014417	8		23	1.7288131	0.0419
facilité	4.3165468	15.00000	5.242464	6		40	-0.2936771	0.3845
passion	0.7194245	2.00000	6.553080	1		50	-3.3626190	0.0004

\$M	% Term/Level	% Level/Term	Global %	Level	Global	occ.	t value	Prob.
intérêt	31.250000	82.27848	31.061599	195		237	0.1288469	0.4487
apprendre	13.141026	85.41667	12.581913	82		96	0.8387528	0.2008
programmation	12.980769	90.00000	11.795544	81		90	2.0974495	0.0180
carrière	11.378205	76.34409	12.188729	71		93	-1.2932207	0.0980
impact	8.653846	77.14286	9.174312	54		70	-0.8990844	0.1843
passion	7.852564	98.00000	6.553080	49		50	3.3626190	0.0004
avenir	6.891026	67.18750	8.387942	43		64	-2.8184192	0.0024
facilité	5.448718	85.00000	5.242464	34		40	0.2936771	0.3845
défaut	2.403846	65.21739	3.014417	15		23	-1.7288131	0.0419

Une fille sur les 92 a exprimé de la passion pour la filière informatique et près de 10% d'entre elles (9), sont intéressées par la programmation.

La Figure VI.7 montre que 70% d'étudiants d'INGMI ayant donné l'intérêt comme raison, ont réussi le cours d'introduction à la programmation et 83.3% pour ceux ayant mentionné une raison liée à la programmation. Ceux qui ont exprimé une raison liée à la facilité ont tous réussi, comme trois étudiants sur les quatre qui ont mentionné la passion.

Figure VI.7. Proportion des raisons données par les étudiants d'INGMI selon leurs notes d'examen

\$`0`	% Term/Level	% Level/Term	Global %	Level	Global	occ.	t value	Prob.
intérêt	30.487805	29.76190	28.474576	25	84	0.33939573	0.3672	
impact	15.853659	27.65957	15.932203	13	47	0.17211092	0.5683	
carrière	14.634146	28.57143	14.237288	12	42	-0.04488481	0.5179	
apprendre	12.195122	28.57143	11.864407	10	35	-0.06881101	0.5274	
avenir	12.195122	29.41176	11.525424	10	34	0.04352537	0.4826	
défaut	7.317073	37.50000	5.423729	6	16	0.62695529	0.2653	
programmation	6.097561	16.66667	10.169492	5	30	-1.23690210	0.1081	
passion	1.219512	25.00000	1.355932	1	4	0.49815557	0.6908	
facilité	0.000000	0.00000	1.016949	0	3	-0.31879947	0.3749	
\$`1`								
intérêt	27.699531	70.23810	28.474576	59	84	-0.33939573	0.3672	
impact	15.962441	72.34043	15.932203	34	47	-0.17211092	0.5683	
carrière	14.084507	71.42857	14.237288	30	42	0.04488481	0.5179	
apprendre	11.737089	71.42857	11.864407	25	35	0.06881101	0.5274	
programmation	11.737089	83.33333	10.169492	25	30	1.23690210	0.1081	
avenir	11.267606	70.58824	11.525424	24	34	-0.04352537	0.4826	
défaut	4.694836	62.50000	5.423729	10	16	-0.62695529	0.2653	
facilité	1.408451	100.00000	1.016949	3	3	0.31879947	0.3749	
passion	1.408451	75.00000	1.355932	3	4	-0.49815557	0.6908	

La Figure VI.8 montre un taux de réussite de 71.7% pour les étudiants d'INFO ayant exprimé un intérêt pour la programmation dans les raisons de leurs choix. 75.7% de ceux qui ont évoqué la « facilité » dans les raisons, ont aussi réussi le cours d'introduction à la programmation.

Figure VI.8. Proportion des raisons données par les étudiants d'INFO selon leurs notes d'examen

\$`0`	% Term/Level	% Level/Term	Global %	Level	Global	occ.	t value	Prob.
intérêt	33.557047	32.67974	32.692308	50	153	0.171050374	0.4321	
apprendre	12.080537	29.50820	13.034188	18	61	-0.258736226	0.3979	
carrière	11.409396	33.33333	10.897436	17	51	0.098687897	0.4607	
programmation	11.409396	28.33333	12.820513	17	60	-0.464683686	0.3211	
passion	10.738255	34.78261	9.829060	16	46	0.299478408	0.3823	
avenir	6.711409	33.33333	6.410256	10	30	0.001043625	0.4996	
facilité	6.040268	24.32432	7.905983	9	37	-0.832197457	0.2026	
impact	6.040268	39.13043	4.914530	9	23	0.557219057	0.2887	
défaut	2.013423	42.85714	1.495726	3	7	0.267099449	0.3947	
\$`1`	% Term/Level	% Level/Term	Global %	Level	Global	occ.	t value	Prob.
intérêt	32.288401	67.32026	32.692308	103	153	-0.171050374	0.4321	
apprendre	13.479624	70.49180	13.034188	43	61	0.258736226	0.3979	
programmation	13.479624	71.66667	12.820513	43	60	0.464683686	0.3211	
carrière	10.658307	66.66667	10.897436	34	51	-0.098687897	0.4607	
passion	9.404389	65.21739	9.829060	30	46	-0.299478408	0.3823	
facilité	8.777429	75.67568	7.905983	28	37	0.832197457	0.2026	
avenir	6.269592	66.66667	6.410256	20	30	-0.001043625	0.4996	
impact	4.388715	60.86957	4.914530	14	23	-0.557219057	0.2887	
défaut	1.253918	57.14286	1.495726	4	7	-0.267099449	0.3947	

Concernant les autres variables liées à la motivation, la répartition montre que près de la moitié d'étudiants d'INFO (135 sur 288, soit 47%) ont exprimé une autodétermination jugée élevée. Du côté d'INGMI, deux tiers (soit 134 sur 201), ont montré une autodétermination jugée moyenne.

Seulement 5.5% d'étudiants d'INGMI ont été jugés avoir un niveau d'abstraction élevé, contre 11% pour les étudiants d'INFO.

Quant à l'apprentissage en autodidacte d'au moins un langage de programmation, il apparaît que 7% d'étudiants d'INGMI (soit 14 sur 201) en étaient intéressés comme 33% (soit 95/288) d'étudiants d'INFO. Les cours préparatoires, quant à eux, ont attiré 88 étudiants d'INGMI et 111 étudiants d'INFO, soit respectivement 43,7% d'étudiants d'INGMI et 38,5% d'étudiants d'INFO.

Tableau VI.28. Répartition du nombre d'étudiants par variable liée à la motivation

Variable	INGMI			INFO		
	Filles	Garçons	Total	Filles	Garçons	Total
autodétermination élevée	10	32	42	8	127	135
autodétermination moyenne	47	87	134	15	120	135
Autodétermination nulle	7	18	25	5	13	18
abstraction élevée	5	6	11	1	32	33

<i>abstraction moyenne</i>	27	58	85	11	119	130
<i>abstraction nulle</i>	32	73	105	16	109	125
<i>autodidacte_prog</i>	2	12	14	8	87	95
<i>cours_prepa</i>	31	57	88	16	95	111

Concernant le nombre de réussites et d'échecs de la population INGMI, le Tableau VI.29 montre que 70% (62/88) de ceux qui ont participé aux cours préparatoires ont réussi le cours d'introduction à la programmation, comme les 78.5% (11/14) de ceux qui ont appris seul au moins un langage de programmation.

Tableau VI.29. Nombre de réussite et d'échec pour les variables liées à la motivation (INGMI)

Réponses	Oui		Non	
	Réussite	Échec	Réussite	Échec
<i>autodidacte_prog</i>	11	3	126	61
<i>cours_prepa</i>	62	26	75	38

88% des étudiants d'INGMI pour lesquels a été trouvée une autodétermination élevée ont réussi (37/42).

Tableau VI.30. Nombre de réussite et d'échec pour l'autodétermination (INGMI)

Valeurs	0 (sans)		1 (moyenne)		2 (élevée)	
	Réussite	Échec	Réussite	Échec	Réussite	Échec
<i>autodétermination</i>	13	12	87	47	37	5

Les étudiants d'INGMI dont les raisons exprimées ont été évaluées par un niveau d'abstraction moyen ou élevé ont un taux de réussite de plus de 70% au cours d'introduction à la programmation. Un taux de réussite de 65.7% (69/105) pour ceux dont le niveau d'abstraction a été évalué comme nul.

Tableau VI.31. Nombre de réussite et d'échec pour le niveau d'abstraction (INGMI)

Valeurs	0 (sans)		2 (moyenne)		4 (élevée)	
	Réussite	Échec	Réussite	Échec	Réussite	Échec
<i>abstraction</i>	69	36	60	25	8	3

Pour la population INFO, les variables « autodidacte\_prog » et « cours\_prepa » dans le Tableau VI.32 montrent respectivement que 82% des étudiants ayant appris seuls la programmation et 75.7% de ceux qui ont participé aux cours préparatoires ont réussi le cours d'introduction à la programmation.

Tableau VI.32. Nombre de réussite et d'échec pour les variables liées à la motivation (INFO)

Réponses	Oui		Non	
	Réussite	Échec	Réussite	Échec
<i>autodidacte_prog</i>	78	17	115	78
<i>cours_prepa</i>	84	27	109	68

Le tableau ci-dessous montre que 70% des étudiants d'INFO ayant une autodétermination élevée ont réussi le cours d'introduction à la programmation.

Tableau VI.33. Nombre de réussite et d'échec pour l'autodétermination (INFO)

Valeurs	0 (sans)		1 (moyenne)		2 (élevée)	
	Réussite	Échec	Réussite	Échec	Réussite	Échec
<i>autodétermination</i>	10	8	88	47	95	40

Quant à l'abstraction, le taux de réussite (61%) pour les étudiants ayant un niveau d'abstraction élevé est inférieur que celui des étudiants ayant un niveau d'abstraction nul (68%) ou moyen (67.7%).

Tableau VI.34. Nombre de réussite et d'échec pour le niveau d'abstraction (INFO)

Valeurs	0 (sans)		2 (moyenne)		4 (élevée)	
	Réussite	Échec	Réussite	Échec	Réussite	Échec
<i>abstraction</i>	85	40	88	42	20	13

## VII. Analyse et Résultats

### VII.1. Choix du modèle

Pour répondre à la question de recherche et identifier les facteurs (les variables) qui favorisent la réussite ou l'échec du cours d'introduction à la programmation en première année INGMI et INFO à l'université de Namur et déterminer les différents profils d'étudiants, il a été choisi de travailler avec les modèles de la régression logistique et l'arbre de décision.

Pourquoi avoir choisi la régression logistique et l'arbre de décision?

Pour répondre à cette question, il est important de connaître les critères aidant à faire ce type de choix<sup>8</sup>:

- **Quantité des données**

Il existe des modèles qui ne seraient pas adaptés pour des dizaines des milliers des données et d'autres si. Près de 10000 réponses ont été traitées pour ce travail Il semblerait que La régression logistique et l'arbre de décision répondent à ce critère selon la figure VI.6.

- **Structure des données**

Les données récoltées sont structurées. Les deux modèles choisis répondent parfaitement à ce critère.

- **Normalité des données**

Il existe des *modèles paramétriques* ayant un nombre de paramètres finis et définis à l'avance tels que la régression linéaire et la régression logistique ; et les modèles non paramétriques pouvant avoir un nombre de paramètres infinis et inconnus à l'avance. Le nombre de variables dans cette étude est un nombre fini. La régression logistique répond à ce critère. L'arbre à décision, comme le montre la figure VI.4, serait utilisé pour des données non paramétriques mais aussi paramétriques.

- **Variables quantitatives et qualitatives**

La régression linéaire, par exemple, n'est adaptée que pour des variables quantitatives. Les variables utilisées pour ce travail sont quasi toutes qualitatives et la figure ci – dessous nous montre que l'arbre à décision répond à ce critère. Quant à la régression logistique, de par sa définition<sup>9</sup>, elle y répond également.

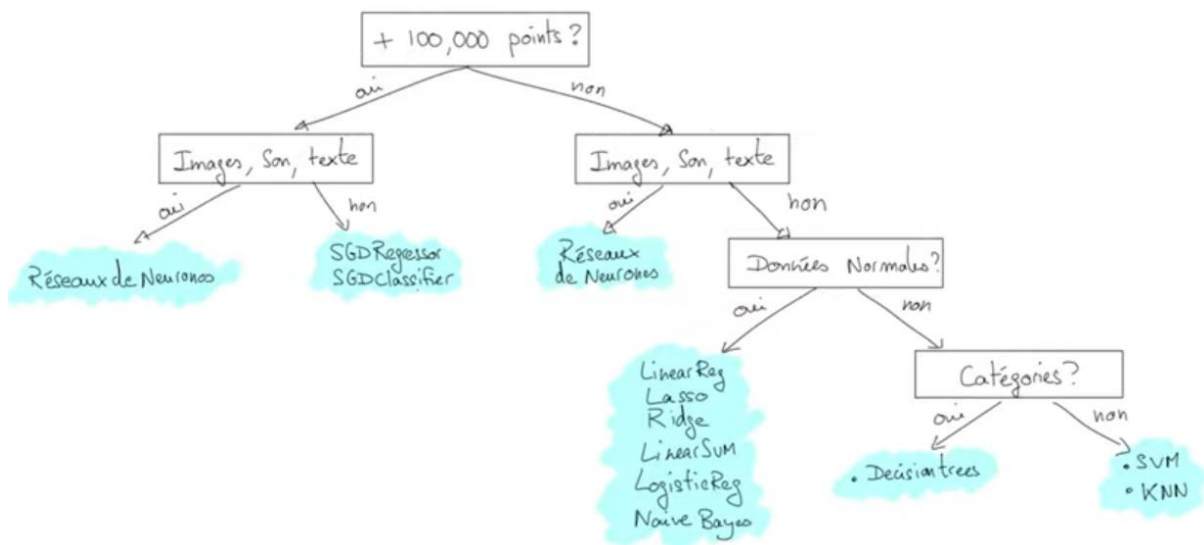
---

<sup>8</sup> Source : <https://www.youtube.com/watch?v=4mqKmTbAnHY>

<sup>9</sup> La régression logistique associe « n » variables indépendantes ( $x_1, \dots, x_n$ ) aléatoires à une variable **binomiale** de type catégorielle « y ».



Figure VII.1. Critère de choix d'un modèle<sup>10</sup>



## VII.2. Méthodes d'analyse

Le but de l'analyse est de lister tous les facteurs qui influencent les notes des étudiants, qu'ils soient en INGMi ou en INFO. Deux types d'analyses ont permis d'étudier l'influence des variables indépendantes :

- Une analyse bivariée entre chaque variable indépendante de notre modèle avec la variable dépendante catégorielle (la note du cours d'introduction à la programmation).
- Une analyse multivariée « explicative » dans laquelle il s'agit d'une part d'expliquer la réussite ou l'échec des étudiants sur base d'un modèle défini par la présence (ou l'absence) de différentes variables indépendantes. Ces dernières seront évaluées via trois méthodes :
  - La déviance résiduelle « **DR** » d'un modèle de régression logistique → **Méthode « DR »**
  - Le critère d'information d'Akaike « **AIC** » (Régression Logistique) → **Méthode « AIC »**
  - L'arbre de décision

Brièvement, la déviance résiduelle (DR), sans entrer dans les termes purement statistiques, est une valeur qui aide à montrer lesquels des modèles saturés ou proposés, expliquent assez bien les données [TeresaStat, 2014]. Un modèle saturé est un modèle qui

<sup>10</sup> Source : <https://www.youtube.com/watch?v=4mqKmTbAnHY>



considère toutes les variables pour évaluer le modèle, pour expliquer, dans le cas de ce mémoire, la note des étudiants. Toutes les variables sont considérées pour voir lesquelles sont significatives pour expliquer la note des étudiants. Un modèle proposé suppose que l'on peut expliquer la note des étudiants avec un nombre de variables inférieur au nombre total des variables. Pour chaque modèle proposé, celui qui donne une valeur de déviance résiduelle plus petit, c'est celui qui explique mieux la valeur dépendante, qui est la note des étudiants au cours d'introduction à la programmation.

Le **critère d'information d'Akaike** (AIC) est utilisé pour *pénaliser les modèles ayant trop de variables*, qui apprennent de trop et généralisent mal<sup>11</sup>. Le but de l'AIC est donc de décrire les données avec le plus petit nombre des variables possibles. Le modèle à retenir est celui ayant l'AIC le plus faible.

L'**arbre de décision** permet de déterminer les différents profils d'étudiants. Il montre les étapes c'est-à-dire les variables par lesquelles passer pour arriver à un résultat qui aboutit soit à un taux de réussite à 100% (ou supérieur à 80%), soit à un taux d'échec à 100% (ou supérieur à 80%) en commençant par la variable racine.

## VII.2.1. Analyse Bivariée

Les objectifs d'une analyse bivariée sont de<sup>12</sup> :

- Vérifier s'il y a un lien entre deux variables d'un échantillon
- Quantifier la force de cette relation si elle existe
- Vérifier enfin si ce lien observé sur l'échantillon est aussi valide au niveau de l'ensemble des données.

Pour ce mémoire, pour chaque variable, il a été déterminé si une relation existe entre cette variable et la « note » des étudiants à l'examen du cours d'introduction à la programmation. Un des éléments indiquant que deux ou plusieurs variables sont liées est la « p-value ». Cette valeur précise que le résultat est statistiquement significatif ou pas selon des seuils de significativité :

$p \leq 0,01$	→ Très forte significativité
$0,01 < p \leq 0,05$	→ Forte significativité
$0,05 < p \leq 0,1$	→ Significativité moyenne
$p > 0,1$	→ Pas de significativité

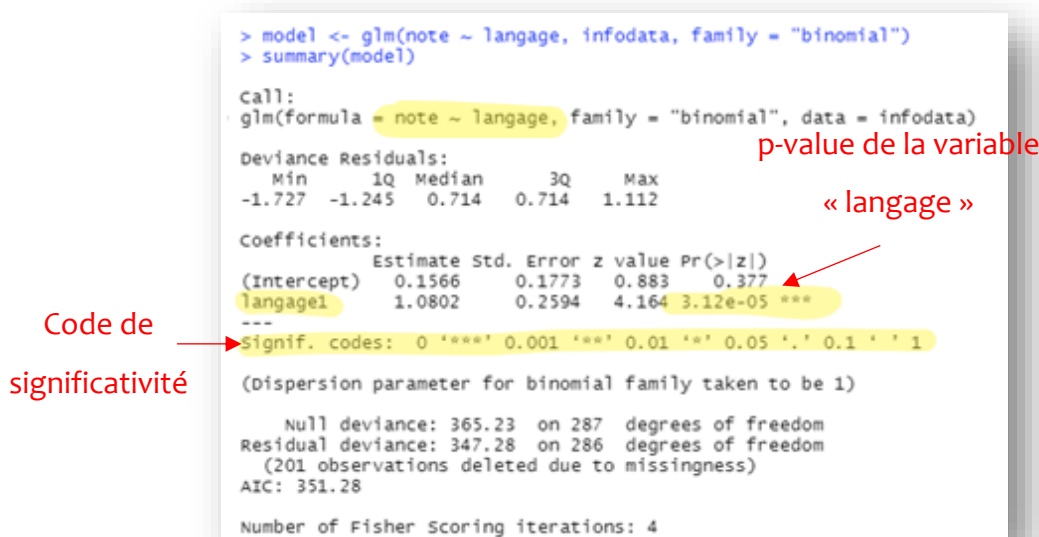
<sup>11</sup> Source : <http://www.jybaudot.fr/Stats/aic.html>

<sup>12</sup> Source : [https://baripedia.org/wiki/Analyse\\_bivari%C3%A9e](https://baripedia.org/wiki/Analyse_bivari%C3%A9e)

La plupart du temps, une variable est significative lorsque la p-value est inférieure à 0.05. Dans le cadre de ce mémoire, toutes les p-values inférieures ou égales à 0.1 seront considérées significatives parce que les conséquences que peuvent avoir les résultats des analyses sur la vie des étudiants sont très négligeables. Ceci n'aurait pas été le cas dans le milieu médical où la santé des personnes est mise en jeu.

En faisant pour chaque variable indépendante une analyse bivariée pour étudier le lien avec la note de l'étudiant. Le résultat obtenu est le suivant :

Figure VII.2. Exemple d'analyse bivariée entre la VD<sup>13</sup> « note » et la variable « langage » (INFO)



La relation entre les codes de significativités dans R et la p-value :

***	$p \leq 0,01$	→	99,9% significatif
**	$0,01 < p \leq 0,05$	→	99% significatif
*	$0,05 < p \leq 0,1$	→	95% significatif
.	$p > 0,1$	→	90% significatif

<sup>13</sup> VD : Variable Dépendante

Un exemple d'analyse bivariée entre la variable « langage » et la note des étudiants d'INFO est présenté sur la figure VI.5. La p-value ( $3.12e-05$ ) suivie de « \*\*\* » montre que le langage est à 99,9% significatif dans ce modèle.

Le deuxième exemple ci-dessous (cf. Figure VI.7), la p-value pour le cours d'algorithme (INFO) est supérieure à 0.1, ce qui montre que ce cours n'influence pas significativement la note de l'étudiant.

Figure VII.3. Exemple Modèle bivarié entre la VD « note » et la variable « cours\_algo » (INFO)

```
> model <- glm(note ~ cours_algo, infodata, family = "binomial")
> summary(model)

Call:
glm(formula = note ~ cours_algo, family = "binomial", data = infodata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6985  -1.4452   0.9314   0.9314   0.9314

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.6106     0.1372   4.451 8.55e-06 ***
cours_algoOui  0.5622     0.3458   1.626  0.104
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 365.23  on 287  degrees of freedom
Residual deviance: 362.42  on 286  degrees of freedom
(201 observations deleted due to missingness)
AIC: 366.42

Number of Fisher Scoring iterations: 4
```

## Analyse bivariée de la population INGMI

Le Tableau VII.1 montre les résultats des analyses bivariées de chaque variable indépendante avec la note des étudiants au cours d'introduction à la programmation. Les p-values ainsi que les codes de significativité de ces analyses montrent la significativité à 90% de la variable exprimant les étudiants ayant eu le cours de bases de données au secondaire « *cours\_db = 'Oui'* », avec la note de l'étudiant et la significativité à 99% de la variable, représentant les étudiants ayant une autodétermination élevée « *autodetermination = 2* » avec la note du cours d'introduction à la programmation.

Tableau VII.1. P-values - Statistiques Bivariées (INGMI)

Variables	p-value	Code de significativité	Significativité
<i>SexeM</i>	0.15		
<i>Statut1</i>	0.549		
<i>Statut2</i>	0.976		

<i>Statut</i> <b>3</b>	0.988		
<i>Cours_info_sec</i> = 'Oui'	0.188		
<i>Bg_info_56</i> = 'Oui'	0.173		
<i>Cours_codage_info</i> = 'Oui'	0.955		
<i>Cours_ordi_arch</i> = 'Oui'	0.108		
<i>Cours_os</i> = 'Oui'	0.528		
<i>Cours_reseau</i> = 'Oui'	0.909		
<i>Cours_html</i> = 'Oui'	0.571		
<i>Cours_algo</i> = 'Oui'	0.984		
<i>Cours_programmation</i> = 'Oui'	0.272		
<i>Cours_db</i> = 'Oui'	0.0874	•	90%
<i>Autodidacte_prog</i> = 'Oui'	0.392		
<i>Langage</i> = <b>1</b>	0.511		
<i>Python</i> = <b>1</b>	0.932		
<i>Cours_prepa</i> = 'Oui'	0.537		
<i>Autodetermination</i> = <b>1</b>	0.222		
<i>Autodetermination</i> = <b>2</b>	0.002	**	99%
<i>Abstraction</i> = <b>2</b>	0.474		
<i>Abstraction</i> = <b>4</b>	0.640		

### Analyse bivariée de la population INFO

L'analyse bivariée de la population INFO montre que le fait d'avoir étudié la programmation au secondaire « *cours\_programmation* = 'Oui' » est significatif à 95% au modèle découlant de l'analyse bivariée entre la variable « *cours\_programmation* » et la note des étudiants au cours d'introduction à la programmation. Même constat pour l'analyse bivariée de la variable « *cours\_prepa* = 'Oui' » (significativité à 95%) avec la note au cours d'introduction à la programmation, celle de la variable « *autodidacte\_prog* = 'Oui' » (significativité à 99.9%), celle de la variable « *langage* = **1** » (significativité à 99.9%), celle de la variable « *python* = **1** » (significativité à 95%) avec la note des étudiants au cours d'introduction à la programmation.

Tableau VII.2. P-values - Statistiques Bivariées (INFO)

Variables	p-value	Codes de significativité	Significativité
<i>Sexe</i> = 'M'	0.116		
<i>Statut</i> = <b>1</b>	0.315		
<i>Statut</i> = <b>2</b>	0.735		
<i>Statut</i> = <b>3</b>	0.409		
<i>Cours_info_sec</i> = 'Oui'	0.717		
<i>Bg_info_56</i> = 'Oui'	0.145		
<i>Cours_codage_info</i> = 'Oui'	0.142		
<i>Cours_ordi_arch</i> = 'Oui'	0.75		
<i>Cours_os</i> = 'Oui'	0.97		
<i>Cours_reseau</i> = 'Oui'	0.61		
<i>Cours_html</i> = 'Oui'	0.171		

<i>Cours_algo</i> = 'Oui'	0.104		
<i>Cours_programmation</i> = 'Oui'	0.0492	*	95%
<i>Cours_db</i> = 'Oui'	0.168		
<i>Autodidacte_prog</i> = 'Oui'	0.00020	***	99.9%
<i>Langage</i> = 1	3.12e-05	***	99.9%
<i>Python</i> = 1	0.0462	*	95%
<i>Cours_prepa</i> = 'Oui'	0.014	*	95%
<i>Autodetermination</i> = 1	0.426		
<i>Autodetermination</i> = 2	0.209		
<i>Abstraction</i> = 2	0.958		
<i>Abstraction</i> = 4	0.425		

## VII.2.2. Analyse multivariée

L'analyse multivariée consiste à considérer toutes les variables indépendantes dans le modèle. Le but est donc de déterminer la meilleure combinaison des variables indépendantes qui explique la note des étudiants. Trois méthodes permettent de déterminer les variables influençant la note des étudiants :

### 1. Méthode – DR

Via une commande pour évaluer le modèle dans R, on peut voir la valeur de la déviance résiduelle (DR).

Figure VII.4. Exemple de modèle multivarié avec affichage de la DR

```
call:
glm(formula = note ~ autodetermination, family = "binomial",
    data = ingmidata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0631  -1.4475   0.5035   0.9294   1.1436

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.08004    0.40032   0.200  0.84152
autodetermination1 0.53572    0.43935   1.219  0.22271
autodetermination2 1.92144    0.62232   3.088  0.00202 **
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 251.52  on 200  degrees of freedom
Residual deviance: 238.92  on 198  degrees of freedom
AIC: 244.92

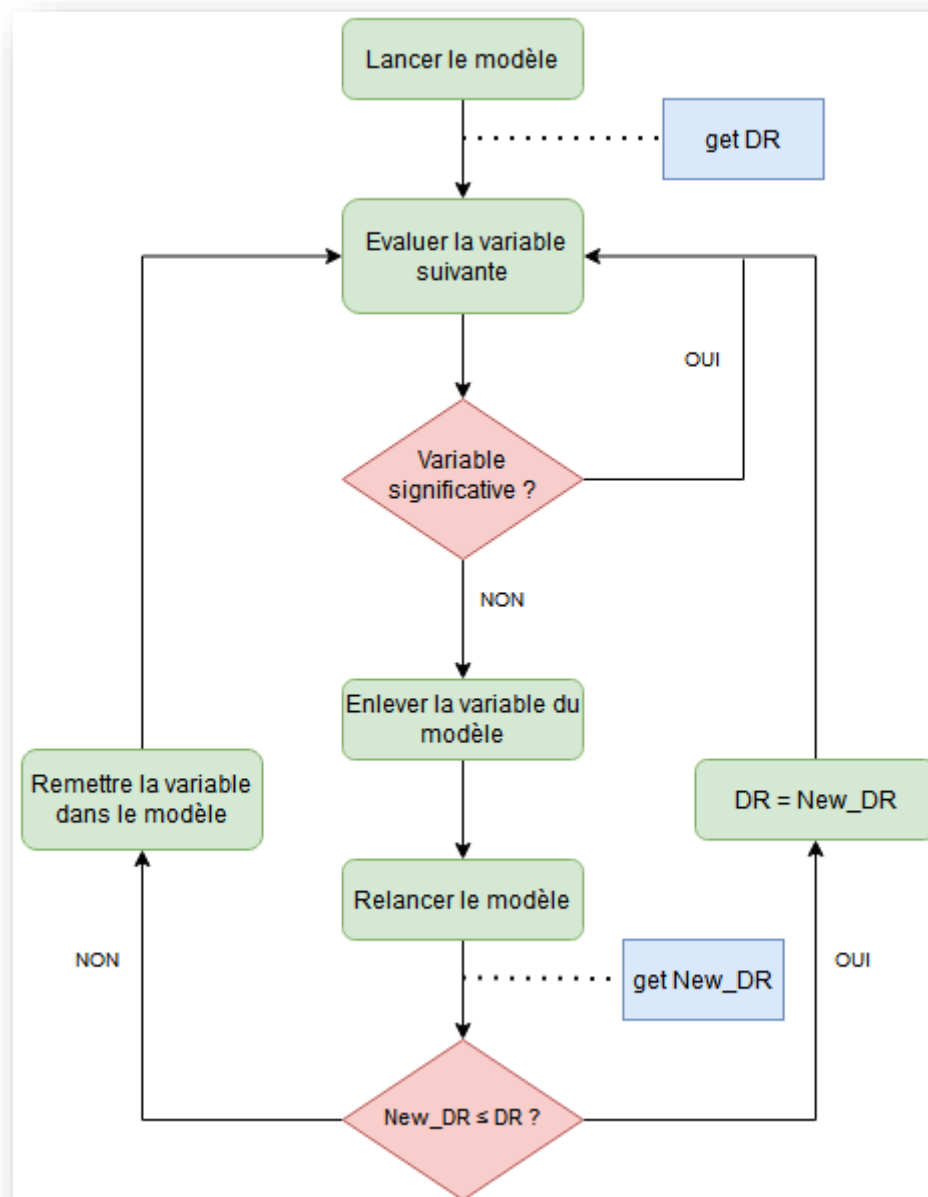
Number of Fisher scoring iterations: 4
```

Comment déterminer quelle variable indépendante garder et laquelle enlever ?

- Le modèle est lancé et la valeur de la déviance résiduelle (**DR**) est extraite du résultat.
- La première variable est évaluée c'est-à-dire que son influence sur le modèle est considérée comme significative ou pas ( $p\text{-value} \leq 0.1$ ).
  - Si oui, elle influence le modèle, la variable suivante est évaluée
  - Sinon, la variable est enlevée et le modèle est relancé → une nouvelle déviance résiduelle « **New\_DR** » est obtenue.
    - Le test  $\text{New\_DR} \leq \text{DR}$  est évalué :
      - Si c'est vrai, cela veut dire que la variable en question n'influence pas significativement le modèle. Elle n'a pas d'importance dans le modèle →  $\text{DR} = \text{New\_DR}$ .
      - Sinon, la variable est remise dans le modèle car elle influence le modèle d'une manière ou d'une autre.

Le flux ci-dessous résume la procédure :

Figure VII.5. Flux pour choisir les variables indépendantes du modèle expliquant la variable indépendante



**Exemple pour la population INGMI :**

**Étape 1 :** Le modèle est exécuté : DR = 201.96

Figure VII.6. Analyse multivariée avec toutes les variables – Méthode DR (INGMI)

```
Call:
glm(formula = note ~ sexe + statut + bg_info_56 + cours_info_sec +
    cours_codage_info + cours_ordi_arch + cours_os + cours_reseau +
    cours_html + cours_db + cours_algo + cours_programmation +
    autodidacte_prog + langage + python + cours_prepa + autodetermination +
    abstraction, family = "binomial", data = ingmidata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0293  -1.0614   0.5436   0.9346   1.3575

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.046e-01  5.578e-01   0.546  0.58507
sexeM          -7.186e-01  3.888e-01  -1.848  0.06460 .
statut1        -1.178e+00  1.825e+00  -0.645  0.51872
statut2        -3.040e-02  7.988e-01  -0.038  0.96964
statut3         1.896e+01  4.449e+03   0.004  0.99660
bg_info_56oui   3.542e+01  2.145e+04   0.002  0.99868
cours_info_secoui 1.122e+01  2.049e+04   0.001  0.99956
cours_codage_infooui 6.532e+01  5.105e+03   0.013  0.98979
cours_ordi_archoui -6.505e+01  4.379e+03  -0.015  0.98815
cours_osoui     -6.313e+01  3.973e+03  -0.016  0.98732
cours_reseauoui  6.880e+01  2.067e+04   0.003  0.99734
cours_htmloui   3.139e+01  2.364e+03   0.013  0.98940
cours_dboui     -9.449e+01  2.070e+04  -0.005  0.99636
cours_algooui   -9.832e+01  9.077e+03  -0.011  0.99136
cours_programmationoui 2.904e+01  2.108e+04   0.001  0.99890
autodidacte_progoui -6.757e-01  1.309e+00  -0.516  0.60565
langage1        8.187e-01  1.229e+00   0.666  0.50517
python1         1.178e+00  1.598e+00   0.737  0.46130
cours_prepaoui   1.348e-01  3.583e-01   0.376  0.70664
autodetermination1 6.316e-01  5.218e-01   1.210  0.22610
autodetermination2 1.973e+00  6.721e-01   2.936  0.00332 **
abstraction2     2.496e-01  3.809e-01   0.655  0.51221
abstraction4    -3.201e-02  8.043e-01  -0.040  0.96825
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 251.52  on 200  degrees of freedom
Residual deviance: 201.96  on 178  degrees of freedom
AIC: 247.96

Number of Fisher Scoring iterations: 18
```

**Étape 2 :** La variable « *sexe* » ne peut pas être enlevée car elle est significative. La variable suivante « *statut* » est enlevée



Figure VII.7. Analyse multivariée sans la variable "statut" - Méthode DR (INGMI)

```
Call:
glm(formula = note ~ sexe - statut + bg_info_56 + cours_info_sec +
  cours_codage_info + cours_ordi_arch + cours_os + cours_reseau +
  cours_html + cours_db + cours_algo + cours_programmation +
  autodidacte_prog + langage + python + cours_prepa + autodetermination +
  abstraction, family = "binomial", data = ingmidata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9558  -1.0865   0.5655   0.9477   1.2969

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.989e-01  5.491e-01   0.727  0.46750
sexeM          -6.754e-01  3.821e-01  -1.768  0.07711 .
bg_info_56Oui   3.476e+01  1.809e+04   0.002  0.99847
cours_info_secOui 1.123e+01  1.750e+04   0.001  0.99949
cours_codage_infoOui 6.476e+01  5.101e+03   0.013  0.98987
cours_ordi_archOui -6.444e+01  4.381e+03  -0.015  0.98827
cours_osOui     -6.267e+01  3.963e+03  -0.016  0.98738
cours_reseauOui  6.823e+01  1.762e+04   0.004  0.99691
cours_htmlOui   3.091e+01  2.353e+03   0.013  0.98952
cours_dbOui     -9.388e+01  1.790e+04  -0.005  0.99581
cours_algoOui   -9.856e+01  8.899e+03  -0.011  0.99116
cours_programmationOui 2.973e+01  1.779e+04   0.002  0.99867
autodidacte_progOui -3.290e-01  1.140e+00  -0.289  0.77293
langage1        7.558e-01  1.046e+00   0.723  0.46976
python1         5.365e-01  1.287e+00   0.417  0.67684
cours_prepaOui  5.889e-02  3.469e-01   0.170  0.86521
autodetermination1 5.979e-01  5.086e-01   1.176  0.23975
autodetermination2 1.841e+00  6.548e-01   2.812  0.00493 **
abstraction2     1.874e-01  3.727e-01   0.503  0.61500
abstraction4     1.424e-02  7.748e-01   0.018  0.98533
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 251.52  on 200  degrees of freedom
Residual deviance: 206.80  on 181  degrees of freedom
AIC: 246.8

Number of Fisher scoring iterations: 18
```

Le New\_DR (206.80) est supérieur au DR de base (201.96). Cela implique que le « statut » a une certaine influence dans ce modèle, la variable est donc remise dans le modèle.

**Étape 3 :** La variable « bg\_info\_56 » est enlevée.

Figure VII.8. Analyse multivariée sans la variable "bg\_info\_56" - Méthode DR (INGMI)

```
call:
glm(formula = note ~ sexe + statut - bg_info_56 + cours_info_sec +
  cours_codage_info + cours_ordi_arch + cours_os + cours_reseau +
  cours_html + cours_db + cours_algo + cours_programmation +
  autodidacte_prog + langage + python + cours_prepa + autodetermination +
  abstraction, family = "binomial", data = ingmidata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0293  -1.0614   0.5436   0.9346   1.3575

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.046e-01  5.578e-01   0.546  0.58507
sexeM          -7.186e-01  3.888e-01  -1.848  0.06460
statut1        -1.178e+00  1.825e+00  -0.645  0.51872
statut2        -3.040e-02  7.988e-01  -0.038  0.96964
statut3         1.995e+01  7.299e+03   0.003  0.99782
cours_info_secOui  1.872e+01  1.511e+04   0.001  0.99901
cours_codage_infoOui  1.057e+02  1.891e+04   0.006  0.99554
cours_ordi_archOui -6.827e+01  7.017e+03  -0.010  0.99224
cours_osOui      -6.626e+01  6.060e+03  -0.011  0.99128
cours_reseauOui   6.555e+01  1.718e+04   0.004  0.99696
cours_htmlOui     3.280e+01  3.368e+03   0.010  0.99223
cours_dbOui       -1.048e+02  1.690e+04  -0.006  0.99505
cours_algoOui     -1.710e+02  1.462e+04  -0.012  0.99067
cours_programmationOui  6.661e+01  6.394e+03   0.010  0.99169
autodidacte_progOui -6.757e-01  1.309e+00  -0.516  0.60565
langage1         8.187e-01  1.229e+00   0.666  0.50517
python1          1.178e+00  1.598e+00   0.737  0.46130
cours_prepaOui    1.348e-01  3.583e-01   0.376  0.70664
autodetermination1  6.316e-01  5.218e-01   1.210  0.22610
autodetermination2  1.973e+00  6.721e-01   2.936  0.00332 **
abstraction2      2.496e-01  3.809e-01   0.655  0.51221
abstraction4     -3.201e-02  8.043e-01  -0.040  0.96825
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 251.52  on 200  degrees of freedom
Residual deviance: 201.96  on 179  degrees of freedom
AIC: 245.96

Number of Fisher Scoring iterations: 19
```

New\_DR = DR implique que la variable « bg\_info\_56 » ne sera pas remise dans le modèle. Elle est donc enlevée définitivement du modèle car ne l'influence en rien. Ce flux est répété jusqu'à ce qu'il n'y ait plus de variable à évaluer.

Le tableau ci-dessous liste les étapes de l'évaluation de chaque variable indépendante statistiquement non significative lors du processus ainsi que les valeurs de la DR respectives.

Tableau VII.3. Étapes d'évaluation du modèle via la DR (INGMI)

DR	Variable enlevée	New_DR	Décision
201.96	statut	206.80	Remettre
201.96	<b>bg_info_56</b>	<b>201.96</b>	<b>Enlever</b>
<u>201.96</u>	cours_info_sec	202.33	Remettre
201.96	cours_codage_info	207.92	Remettre
201.96	cours_ordi_arch	209.77	Remettre
201.96	cours_os	207.68	Remettre
201.96	cours_reseau	203.48	Remettre
201.96	cours_html	203.99	Remettre
201.96	cours_db	212.18	Remettre

201.96	cours_algo	215.86	Remettre
201.96	cours_programmation	206.27	Remettre
201.96	autodidacte_prog	202.23	Remettre
201.96	langage	202.43	Remettre
201.96	python	202.56	Remettre
201.96	cours_prepa	202.10	Remettre
201.96	abstraction	202.45	Remettre

## Modèle final pour la population INGMI

Ce modèle final (cf. Figure VII.9) montre que :

- La variable « **autodétermination** » y est significative à 99%. Avoir une autodétermination élevée (**autodétermination = 2**) est un facteur important influençant la réussite au cours d'introduction à la programmation. Ceci était déjà le cas dans l'analyse bivariée.
- La variable « **sexe** » est significative à 90%. Son coefficient estimé ( $-7.119e-01$ ), lorsqu'il s'agit du sexe masculin, est négatif. Le fait d'être un homme en INGMI aurait une influence statistique négative sur la réussite dans ce modèle, et ce malgré que le taux de réussite des hommes en INGMI soit de 65% (89 sur 137).

Figure VII.9. Modèle Final INGMI (Méthode DR)

```

call:
glm(formula = note ~ sexe + statut - bg_info_56 + cours_info_sec +
cours_codage_info + cours_ordi_arch + cours_os + cours_reseau +
cours_html + cours_db + cours_algo + cours_programmation +
autodidacte_prog + langage + python + cours_prepa + autodetermination +
abstraction, family = "binomial", data = ingmidata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0293  -1.0614   0.5436   0.9346   1.3575

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.046e-01  5.578e-01   0.546  0.58507
sexeM          -7.186e-01  3.888e-01  -1.848  0.06460 .
statut1        -1.178e+00  1.825e+00  -0.645  0.51872
statut2        -3.040e-02  7.988e-01  -0.038  0.96964
statut3         1.995e+01  7.299e+03   0.003  0.99782
cours_info_secoui 1.872e+01  1.511e+04   0.001  0.99901
cours_codage_infooui 1.057e+02  1.891e+04   0.006  0.99554
cours_ordi_archoui -6.827e+01  7.017e+03  -0.010  0.99224
cours_osoui      -6.626e+01  6.060e+03  -0.011  0.99128
cours_reseauoui  6.555e+01  1.718e+04   0.004  0.99696
cours_htmloui    3.280e+01  3.368e+03   0.010  0.99223
cours_dboui      -1.048e+02  1.690e+04  -0.006  0.99505
cours_algooui    -1.710e+02  1.462e+04  -0.012  0.99067
cours_programmationoui 6.661e+01  6.394e+03   0.010  0.99169
autodidacte_progoui -6.757e-01  1.309e+00  -0.516  0.60565
langage1         8.187e-01  1.229e+00   0.666  0.50517
python1          1.178e+00  1.598e+00   0.737  0.46130
cours_prepaoui   1.348e-01  3.583e-01   0.376  0.70664
autodetermination1 6.316e-01  5.218e-01   1.210  0.22610
autodetermination2 1.973e+00  6.721e-01   2.936  0.00332 **
abstraction2      2.496e-01  3.809e-01   0.655  0.51221
abstraction4     -3.201e-02  8.043e-01  -0.040  0.96825
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 251.52  on 200  degrees of freedom
Residual deviance: 201.96  on 179  degrees of freedom
AIC: 245.96

Number of Fisher Scoring iterations: 19

```

## Modèle final pour la population INFO

La DR de départ était 323.62. La première variable à être enlevée était le « sexe » mais cela n'a pas amélioré le modèle car la nouvelle valeur de la DR (New\_DR = 325.13 > 323.62 → pas OK) montre que la variable « sexe » a une influence sur le modèle.

Le tableau ci-dessous liste les étapes de l'évaluation de chaque variable indépendante statistiquement non significative lors du processus ainsi que les valeurs de la DR et de l'AIC respectives.

Tableau VII.4. Étapes d'évaluation du modèle via la DR et l'AIC (INFO)

DR	Variable enlevée	New_DR	Décision
323.62	sexe	325.13	Remettre
323.62	statut	324.86	Remettre
323.62	bg_info_56	323.81	Remettre
323.62	cours_info_sec	326.05	Remettre
323.62	cours_codage_info	323.67	Remettre
323.62	cours_ordi_arch	323.75	Remettre
323.62	cours_os	324.05	Remettre
323.62	cours_reseau	323.78	Remettre
323.62	cours_html	324.62	Remettre
323.62	cours_db	323.65	Remettre
323.62	cours_algo	323.68	Remettre
323.62	cours_programmation	324.01	Remettre
323.62	python	324.02	Remettre
323.62	autodetermination	324.43	Remettre
323.62	abstraction	325.10	Remettre

La figure VII.10 montre que les variables « *autodidacte\_prog* », « *langage* » et « *cours\_prepa* » influencent significativement le modèle pour la population INFO. Aucune variable n'a pu être enlevée.

Le résultat de ce modèle (voir Figure VII.10) montre donc que :

- La variable « *cours\_prepa* » est significative à 95% dans ce modèle (comme pour l'analyse bivariée) et son coefficient estimé, pour les étudiants ayant répondu « oui », est positif. *Participer aux cours préparatoires*, pour les étudiants d'INFO, aurait une forte influence sur la réussite au cours d'introduction à la programmation selon la méthode DR.
- Les variables « *langage* » et « *autodidacte\_prog* » sont toutes les deux significatives à 90% (comme pour l'analyse bivariée) et leurs coefficients estimés respectifs, pour les étudiants ayant répondu « Oui », sont positifs. Avoir appris un langage de programmation et l'avoir fait en autodidacte aurait une influence sur les notes de l'examen du cours d'introduction à la programmation.

Figure VII.10. Modèle Final INFO (Méthode DR)

```
Call:
glm(formula = note ~ sexe + statut + bg_info_56 + cours_info_sec +
    cours_codage_info + cours_ordi_arch + cours_os + cours_reseau +
    cours_html + cours_db + cours_algo + cours_programmation +
    autodidacte_prog + langage + python + cours_prepa + autodetermination +
    abstraction, family = "binomial", data = infodata)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.3203	-1.0844	0.5904	0.8525	1.5234

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.8689	0.6713	-1.294	0.1956
sexeM	0.5996	0.4863	1.233	0.2176
statut1	-0.6180	0.8694	-0.711	0.4772
statut2	-0.3969	0.5123	-0.775	0.4386
statut3	0.3540	0.9202	0.385	0.7004
bg_info_56oui	0.3266	0.7604	0.430	0.6675
cours_info_secoui	-1.2060	0.7736	-1.559	0.1190
cours_codage_infooui	-0.1545	0.6964	-0.222	0.8244
cours_ordi_archoui	-0.2594	0.7410	-0.350	0.7263
cours_osoui	-0.4659	0.7228	-0.645	0.5192
cours_reseauoui	0.2464	0.6255	0.394	0.6936
cours_htmloui	0.7205	0.7239	0.995	0.3196
cours_dboui	0.1080	0.6547	0.165	0.8689
cours_algooui	0.1623	0.6615	0.245	0.8061
cours_programmationoui	0.5351	0.8636	0.620	0.5355
autodidacte_progoui	0.8053	0.4744	1.697	0.0896 .
langage1	0.8937	0.5317	1.681	0.0928 .
python1	-0.2762	0.4382	-0.630	0.5285
cours_prepaoui	0.7395	0.3037	2.435	0.0149 *
autodetermination1	0.5277	0.5866	0.900	0.3683
autodetermination2	0.4526	0.5914	0.765	0.4441
abstraction2	-0.1767	0.3089	-0.572	0.5672
abstraction4	-0.5674	0.4677	-1.213	0.2251

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 365.23 on 287 degrees of freedom  
Residual deviance: 323.62 on 265 degrees of freedom  
(201 observations deleted due to missingness)  
AIC: 369.62

Number of Fisher Scoring iterations: 4

## 2. Méthode – AIC

Dans cette méthode, c'est l'AIC qui est évalué. La méthode évalue des successions de modèles emboîtés en alternant l'ajout et le retrait d'une ou des variables pour avoir un modèle optimal. Le programme s'arrête quand l'ajout et le retrait d'une variable n'améliore plus le critère. C'est le même procédé que celui mis en place dans la méthode DR sauf que c'est le critère d'information d'Akaike qui est pris en compte. En outre, l'évaluation est faite automatiquement via une commande R à l'inverse de la méthode DR où c'est l'utilisateur qui relance à chaque fois l'évaluation et décide d'enlever ou de garder une variable.

**Exemple pour la population INGMI :**

Figure VII.11. Début Méthode AIC (INGMI)

```

Start: AIC=247.96
note ~ sexe + statut + bg_info_56 + cours_info_sec + cours_codage_info +
      cours_ordi_arch + cours_os + cours_reseau + cours_html +
      cours_db + cours_algo + cours_programmation + autodidacte_prog +
      langage + python + cours_prepa + autodetermination + abstraction

      Df Deviance   AIC
- abstraction      2  202.44 244.44
- cours_programmation 1  201.96 245.96
- bg_info_56        1  201.96 245.96
- cours_reseau      1  201.96 245.96
- cours_info_sec    1  201.96 245.96
- cours_prepa       1  202.10 246.10
- autodidacte_prog  1  202.23 246.23
- langage           1  202.43 246.43
- python            1  202.56 246.56
- statut            3  206.81 246.81
<none>             1  201.96 247.96
- cours_html       1  203.99 247.99
- sexe             1  205.54 249.54
- cours_os         1  206.91 250.91
- cours_codage_info 1  207.92 251.92
- cours_ordi_arch  1  209.76 253.76
- autodetermination 2  212.98 254.98
- cours_db         1  212.12 256.12
- cours_algo       1  214.85 258.85

Step: AIC=244.45
note ~ sexe + statut + bg_info_56 + cours_info_sec + cours_codage_info +
      cours_ordi_arch + cours_os + cours_reseau + cours_html +
      cours_db + cours_algo + cours_programmation + autodidacte_prog +
      langage + python + cours_prepa + autodetermination

```

L'AIC de départ pour cette méthode est de 247.96 et la variable enlevée, « abstraction », est celle avec le plus petit AIC (244.44).

Ensuite l'évaluation est lancée sans la variable « abstraction » dans le modèle et avec comme AIC de départ, le plus petit de l'exécution précédente. La Figure VII.12 affiche la variable « abstraction » avec un « + » pour dire qu'elle n'a pas été utilisée dans le nouveau modèle.

Figure VII.12. Méthode AIC après une première évaluation (INGMI)

```

Step: AIC=244.45
note ~ sexe + statut + bg_info_56 + cours_info_sec + cours_codage_info +
      cours_ordi_arch + cours_os + cours_reseau + cours_html +
      cours_db + cours_algo + cours_programmation + autodidacte_prog +
      langage + python + cours_prepa + autodetermination

      Df Deviance   AIC
- cours_programmation 1 202.44 242.44
- bg_info_56           1 202.44 242.44
- cours_reseau         1 202.44 242.44
- cours_info_sec       1 202.44 242.44
- cours_prepa          1 202.55 242.55
- autodidacte_prog     1 202.68 242.68
- langage              1 202.84 242.84
- statut               3 207.07 243.07
- python               1 203.09 243.09
<none>                 1 202.44 244.44
- cours_html           1 204.57 244.57
- sexe                 1 205.96 245.96
- cours_os             1 207.22 247.22
+ abstraction           2 201.96 247.96
- cours_codage_info    1 208.49 248.49
- cours_ordi_arch      1 210.07 250.07
- autodetermination    2 213.46 251.46
- cours_db             1 212.89 252.89
- cours_algo           1 215.56 255.56

Step: AIC=242.45
note ~ sexe + statut + bg_info_56 + cours_info_sec + cours_codage_info +
      cours_ordi_arch + cours_os + cours_reseau + cours_html +
      cours_db + cours_algo + autodidacte_prog + langage + python +
      cours_prepa + autodetermination

```

Le nouvel AIC de cette étape est de 244.45 et la variable enlevée « *cours\_programmation* » est celle avec le plus petit AIC (242.44).

Le processus se répète jusqu'à ce que, comme dit plus haut, il n'améliore plus le critère.

### Modèle final pour la population INGMI

Au final, les variables indépendantes précédées du signe « - » sur la Figure VII.13 sont celles expliquant la note des étudiants en INGMI selon cette méthode. Le résultat montre qu'il y a huit variables indépendantes en moins dans le modèle par rapport à la méthode utilisant la déviance résiduelle (méthode DR).

Figure VII.13. Fin Méthode AIC (INGMI)

```

Step:  AIC=233.82
note ~ sexe + bg_info_56 + cours_codage_info + cours_ordi_arch +
      cours_os + cours_reseau + cours_html + cours_db + cours_algo +
      autodetermination

```

	Df	Deviance	AIC
<none>		209.82	233.82
+ langage	1	207.91	233.91
- cours_html	1	212.45	234.45
+ python	1	208.64	234.64
+ statut	3	204.80	234.80
+ cours_info_sec	1	208.90	234.90
+ cours_programmation	1	208.90	234.90
- sexe	1	212.99	234.99
+ autodidacte_prog	1	209.27	235.27
+ cours_prepa	1	209.81	235.81
- cours_os	1	214.47	236.47
+ abstraction	2	209.49	237.49
- cours_codage_info	1	216.16	238.16
- bg_info_56	1	217.56	239.56
- autodetermination	2	220.49	240.49
- cours_reseau	1	219.05	241.05
- cours_db	1	220.87	242.87
- cours_ordi_arch	1	221.27	243.27
- cours_algo	1	224.26	246.26

La figure ci-dessous donne le modèle final pour la population INGMI dans lequel les variables « autodétermination » et « sexe » influencent significativement le modèle comme c'était le cas de la méthode « DR ». Malgré la différence du nombre des variables composant les deux modèles, le même résultat c'est-à-dire le même code de significativité pour les mêmes variables, même signe de coefficient estimé que pour la méthode « DR » est obtenu.



Figure VII.14. Modèle Final INGMI (Méthode AIC)

```
call:
glm(formula = note ~ sexe + bg_info_56 + cours_codage_info +
  cours_ordi_arch + cours_os + cours_reseau + cours_html +
  cours_db + cours_algo + autodetermination, family = "binomial",
  data = ingmidata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9584  -1.1034   0.5638   0.9572   1.2533

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      0.4788     0.5129   0.934  0.35049
sexeM            -0.6555     0.3774  -1.737  0.08245 .
bg_info_56Oui     81.3804    6494.9161   0.013  0.99000
cours_codage_infoOui 81.6596    8243.4272   0.010  0.99210
cours_ordi_archOui -99.6903    7882.4287  -0.013  0.98991
cours_osOui       -78.1335    6315.2238  -0.012  0.99013
cours_reseauOui   114.2097    8665.1976   0.013  0.98948
cours_htmlOui     45.9441    4103.3986   0.011  0.99107
cours_dbOui       -112.9806    8452.7167  -0.013  0.98934
cours_algoOui     -102.7069   11573.8778  -0.009  0.99292
autodetermination1  0.7195     0.4735   1.520  0.12863
autodetermination2  1.9355     0.6474   2.990  0.00279 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 251.52  on 200  degrees of freedom
Residual deviance: 209.82  on 189  degrees of freedom
AIC: 233.82

Number of Fisher Scoring iterations: 19
```

## Modèle final pour la population INFO

La méthode AIC pour la population INFO montre que seules cinq variables ont été gardées pour ce modèle. Toutes les variables précédées d'un « + » sur la Figure VII.15 ont été donc enlevées du modèle.

Figure VII.15. Fin Méthode AIC (INFO)

```
Step: AIC=343.21
note ~ cours_info_sec + cours_programmation + autodidacte_prog +
  langage + cours_prepa

              Df Deviance   AIC
<none>                331.21 343.21
+ sexe                 1  329.95 343.95
- autodidacte_prog     1  334.00 344.00
+ cours_html           1  330.08 344.08
- langage              1  334.20 344.20
+ python               1  330.67 344.67
- cours_programmation  1  334.71 344.71
+ bg_info_56           1  330.81 344.81
+ cours_ordi_arch      1  330.92 344.92
+ cours_db             1  330.93 344.93
+ cours_algo           1  331.04 345.04
+ cours_os             1  331.10 345.10
+ cours_reseau         1  331.17 345.17
+ cours_codage_info    1  331.21 345.21
- cours_info_sec       1  336.38 346.38
+ autodetermination    2  330.43 346.43
+ abstraction          2  330.59 346.59
+ statut              3  329.48 347.48
- cours_prepa          1  339.33 349.33
```

Les résultats du modèle donnent ceci :

- La variable « **cours\_info\_sec** » est significative à 95% et son coefficient estimé, pour les étudiants ayant répondu « Oui », est négatif (**-1.07644**). Avoir eu des cours d'informatique à l'école secondaire aurait une influence statistiquement négative sur la réussite des étudiants au cours d'introduction à la programmation.
- La variable « **cours\_programmation** » est significative à 90% et son coefficient estimé, pour les étudiants ayant répondu « Oui », est positif. Les étudiants en INFO ayant eu un cours de programmation à l'école secondaire, auraient plus de chance de réussir au cours d'introduction à la programmation.
- Les variables « **langage** » et « **autodidacte\_prog** » sont toutes les deux significatives à 90% (comme pour la méthode DR) et leurs coefficients estimés respectifs, pour les étudiants ayant répondu « Oui », sont positifs. Avoir appris un langage de programmation et de l'avoir fait en autodidacte, influencerait les notes de l'examen du cours d'introduction à la programmation.
- La variable « **cours\_prepa** » devient<sup>14</sup> significative à 99% dans ce modèle et son coefficient estimé, pour les étudiants ayant répondu « oui », est positif. Participer aux cours préparatoires, pour les étudiants en faculté INFO, influencerait fortement la réussite au cours d'introduction à la programmation selon la méthode AIC.

Figure VII.16. Modèle Final INFO (Méthode AIC)

```
Call:
glm(formula = note ~ cours_info_sec + cours_programmation + autodidacte_prog +
    langage + cours_prepa, family = "binomial", data = infodata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1420  -1.1526   0.6662   0.8846   1.6816

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.05869    0.21726  -0.270   0.78704
cours_info_secOui -1.07644    0.47783  -2.253   0.02427 *
cours_programmationOui 1.08897    0.58443   1.863   0.06242 .
autodidacte_progOui  0.70069    0.42160   1.662   0.09652 .
langage1      0.73813    0.43225   1.708   0.08770 .
cours_prepaOui  0.79511    0.28595   2.781   0.00543 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 365.23  on 287  degrees of freedom
Residual deviance: 331.21  on 282  degrees of freedom
AIC: 343.21

Number of Fisher Scoring iterations: 4
```

<sup>14</sup> Dans méthode DR, la variable était juste « fortement » significative

### 3. Méthode 3 – Arbre de décision

Dans la méthode de l'arbre de décision, le but n'est pas d'enlever les variables au fur et à mesure. *Les variables indépendantes qui ont été retenues, sont celles qui sont plus près de la racine et qui mènent vers un nœud-feuille dont la probabilité de la réussite ou de l'échec est supérieure à 80%.* La racine est toujours significative au modèle représenté.

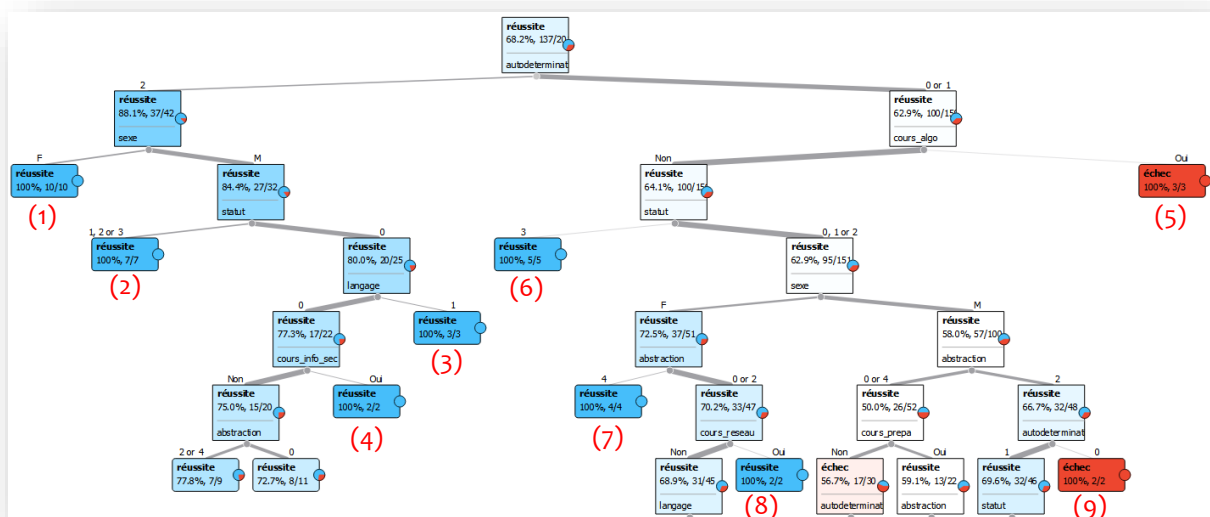
Les nœuds les plus intéressants sont ceux à 100% car ils déterminent précisément les différentes caractéristiques amenant soit à un échec, soit à une réussite. Plus les couleurs sont foncées plus la conclusion est facile.

#### Analyse pour la population INGM

En regardant l'arbre ci-dessous, il peut être déduit que :

- La variable « *autodetermination* » qui est la racine de l'arbre, est significative pour ce modèle.
- Sept des neuf nœuds – feuilles<sup>15</sup> dont la proportion est de 100%, finissent par une réussite et les deux autres par un échec.
- Huit variables sont impliquées dans la formation des nœuds-feuilles: *abstraction* (AB), *autodetermination*(AU), *cours\_algo* (CA), *cours\_info\_sec* (CI), *cours\_reseau* (CR), *langage* (L), *sexe* (SX), *statut* (ST)

Figure VII.17. Arbre de décision INGM (Profondeur = 7)



<sup>15</sup> Nœuds – feuilles : Ce sont des nœuds terminaux qui ne peuvent pas avoir de nœuds-enfants

Le résultat final (cf. Tableau VII.5) montre pour chaque nœud-feuille (numéroté de 1 à 9) représentant le résultat final à 100%, les facteurs (les nœuds) influençant ce résultat. Les facteurs en question sont les variables citées précédemment et impliquées dans la formation des nœuds-feuilles.

Le Tableau VII.5 montre que :

- (1) Une étudiante ayant une autodétermination élevée (***autodétermination = 2***) a 100% de chance de réussir
- (2) Un étudiant de sexe masculin, ayant une autodétermination élevée (***autodétermination = 2***) a 100% de chance de réussir s'il n'est pas nouveau (***statut ≠ 0***).
- (3) Un étudiant de sexe masculin venant directement du secondaire (***statut = 0***), ayant une autodétermination élevée (***autodétermination = 2***), a 100% des chances de réussir s'il a déjà appris un langage de programmation (***langage = 1***)
- (4) Un étudiant de sexe masculin venant directement du secondaire (***statut = 0***), ayant une autodétermination élevée (***autodétermination = 2***), mais n'a jamais appris un langage de programmation (***langage = 0***), a 100% de chance de réussir s'il a déjà eu un cours d'informatique au secondaire (***cours\_info\_sec = Oui***)
- (5) Les étudiants n'ayant pas une autodétermination élevée (***autodétermination ≠ 2***), ont 0% de chance de réussir s'ils ont eu cours d'algorithme (***cours\_algo = 1***) à l'école secondaire.
- (6) Les étudiants n'ayant pas une autodétermination élevée (***autodétermination ≠ 2***) et qui n'ont pas eu de cours d'algorithme (***cours\_algo = 0***) à l'école secondaire, ont 100% de chance de réussir s'ils font partie d'autres cas de statut (***statut = 3***). Les autres (***statut ≠ 3***), s'ils sont de sexe féminin et ont un niveau d'abstraction élevé (***abstraction = 4***), alors ils auront aussi 100% (4 sur 4) de chance de réussir. Sinon (***abstraction ≠ 4***), il faut qu'ils aient eu cours de réseau au secondaire pour avoir 100% de chance de réussir.
- (7) Les étudiantes n'ayant pas une autodétermination élevée (***autodétermination ≠ 2***), qui n'ont pas eu de cours d'algorithme (***cours\_algo = 0***) à l'école secondaire, qui ne font pas partie d'autres cas de statut (***statut ≠ 3***), ont 100% de chance de réussir si elles ont un niveau d'abstraction élevé (***abstraction = 2***)
- (8) Les étudiantes n'ayant pas une autodétermination élevée (***autodétermination ≠ 2***), qui n'ont pas eu de cours d'algorithme (***cours\_algo = 0***) à l'école secondaire, qui ne font pas partie d'autres cas de statut (***statut ≠ 3***) et qui ont un niveau d'abstraction qui n'est pas élevé (***abstraction ≠ 4***), ont 100% de chance de réussir si elles ont eu un cours de réseau (***cours\_reseau = Oui***).

- (9) Les étudiants de sexe masculin, ayant une autodétermination nulle (*autodetermination* = 0), ne faisant pas partie d'autres cas de statut (statut  $\neq$  3), n'ayant pas eu cours d'algorithme au secondaire et dont le niveau d'abstraction est moyen (*abstraction* = 2) n'ont aucune chance de réussir.

Tableau VII.5. Composition des variables amenant vers la réussite ou l'échec (INGMI)

		BI		BP		M		
SX	ST	CI	CR	CA	L	AU	AB	Résultat
F						2		(1)
M	1 2 3					2		(2)
M	0				1	2		(3)
M	0	Oui			0	2		(4)
				Oui		0 1		(5)
	3			Non		0 1		(6)
F	0 1 2			Non		0 1	4	(7)
F	0 1 2		Oui	Non		0 1	0 2	(8)
M	0 1 2			Non		0	2	(9)

**Légende**  
*M* → Variables liées à la motivation  
*BI* → Variables liées au bagage en informatique  
*BP* → Variables liées au bagage en programmation

### Analyse pour la population INFO

En regardant l'arbre ci-dessous, il peut être déduit que :

- La variable « *langage* » qui est la racine de l'arbre, est significative pour ce modèle.
- Cinq des huit nœuds – feuilles numérotés, finissent par une probabilité élevée de réussite et les trois autres se terminent par une probabilité élevée d'échec.
- Huit variables sont impliquées dans la formation des nœuds-feuilles: *abstraction* (AB), *autodetermination*(AU), *cours\_codage\_info* (CC), *cours\_info\_sec*(CI), *cours\_os* (CO), *cours\_prepa* (CP), (CR), *langage* (L), *statut* (ST)

[illegible]

- (1) Les étudiants qui n'ont appris aucun langage de programmation (**langage = 0**) mais qui ont eu le cours de codage de l'information à l'école secondaire, ont 0% de chance de réussir.
- (2) Les étudiants bisseurs (**statut = 1**) ou faisant partie d'autres cas de statut (**statut = 3**), qui n'ont appris aucun langage de programmation, ni n'ont eu le cours de codage de l'information en secondaire, ni participés aux cours préparatoires, ont 100% (3 sur 3) de chance de réussir.
- (3) Les étudiants venant d'une autre filière (**statut = 2**) qui n'ont appris aucun langage de programmation, qui n'ont pas appris non plus le cours de codage de l'information en secondaire, ni participé aux cours préparatoires, ont 100% (3 sur 3) de chance de réussir.
- (4) Les étudiants qui n'ont appris aucun langage de programmation, ni n'ont eu le cours de codage de l'information en secondaire mais qui ont participé aux cours préparatoires et ont eu des cours d'informatique en secondaire, ont 100% de chance de réussir.
- (5) Les étudiants « bisseurs » (statut = 1) ou faisant partie d'autres cas de statut (statut = 3), qui ont appris au moins un langage de programmation (langage = 1), qui n'ont pas suivi les cours préparatoires mais qui ont eu des cours d'informatique à l'école secondaire, ont 0% (0 sur 3) de chance de réussir.
- (6) Les étudiants venant d'une autre filière (statut = 2) qui ont appris au moins un langage de programmation (langage = 1), qui n'ont pas suivi les cours préparatoires mais dont

l'autodétermination n'est pas nulle (autodetermination  $\neq 0$ ), ont 86.7% (13/15) de chances de réussir.

- (7) Les étudiants « nouveaux » (statut = 0) et ceux venant d'une autre filière (statut = 2), qui ont appris au moins un langage de programmation (langage = 1), qui n'ont pas suivi les cours préparatoires et dont l'autodétermination est nulle (autodetermination = 0) ont 100 de chance de réussir
- (8) Les étudiants nouveaux (statut = 0) ou faisant partie d'autres cas de statut (statut = 3), qui ont participé aux cours préparatoires, qui ont un niveau d'abstraction non élevé (abstraction  $\neq 4$ ) et ayant suivi le cours de système d'exploitation en secondaire, ont 96.9% de chance de réussir le cours d'introduction à la programmation.

Tableau VII.6. Composition des variables amenant vers la réussite ou l'échec

	BI			BP	M			
ST	CC	CI	CO	L	AU	AB	CP	Résultat
	Oui			0				(1)
1 3	Non			0			Non	(2)
2	Non			0			Non	(3)
	Non	Oui		0			Oui	(4)
1 3		Oui		1			Non	(5)
2				1	1 2		Non	(6)
0 2				1	0		Non	(7)
0 3			Non	1		0 2	Oui	(8)
<b>Légende</b> <i>M</i> → Variables liées à la motivation <i>BI</i> → Variables liées au bagage en informatique <i>BP</i> → Variables liées au bagage en programmation								

### VII.3. Profil des étudiants en INGMI

#### VII.3.1. Rappel des résultats

##### Taux de réussite

68.2% (137/201)

##### Taux de réussite par sexe

Filles – 75% (48/64)

Garçons – 65% (89/137)

##### Taux de réussite par statut

Nouveaux – 67% (118/176)

Bisseurs – 80% (4/5)

Autres filières – 67% (10/15)

Autres cas – 100% (5/5)

##### Taux de réussite par variables du bagage en informatique

Cours\_info\_sec : **Oui – 55% (11/20)**

Bg\_info\_56 : **Oui – 50% (6/12)**

Cours_info_sec : Non – 69.6% (126/181)	Bg_info_56 : <b>Non – 70.4% (131/186)</b>
Cours_codage_info : Oui – 66.7% (2/3)	Cours_ordi_arch : <b>Oui – 45,4% (5/11)</b>
Cours_codage_info : Non – 68.2% (135/198)	Cours_ordi_arch : Non – 69.5% (132/190)
Cours_os : Oui – 57.1% (4/7)	Cours_reseau : Oui – 66.7% (8/12)
Cours_os : Non – 68.6% (133/194)	Cours_reseau : Non – 68.2% (129/189)
Cours_html : <b>Oui – 80% (4/5)</b>	Cours_db : <b>Oui – 33.3% (2/6)</b>
Cours_html : Non – 67.9% (133/196)	Cours_db : Non – 69.2% (135/195)

### Taux de réussite par variables du bagage en programmation

Cours_programmation	<b>Oui – 50% (4/8)</b>	Non – 69% (133/193)
Cours_algo	<b>Oui – 0% (0/3)</b>	Non – 69.2% (137/198)
Langage	<b>1 – 73.3 (22/30)</b>	0 – 67.2% (115/171)

### Taux de réussite par variables du bagage en Python

Python	1 – 69.2% (9/13)	0 – 68% (128/188)
--------	------------------	-------------------

### Taux de réussite par variables liées à la motivation

Raison_filière	Intérêt – <b>70% (59/84)</b>	Facilité – <b>100% (3/3)</b>
	Passion – <b>75% (3/4)</b>	Programmation – <b>83.3% (25/30)</b>

Autodidacte_prog : <b>Oui – 78.6% (11/14)</b>	Cours_prepa : <b>Oui – 70.4% (62/88)</b>
Autodidacte_prog : Non – 67.4% (126/187)	Cours_prepa : Non – 66.4% (75/113)

Autodétermination	<b>0 – 52% (13/25)</b>	1 – 65% (87/134)	2 – 88% (37/42)
Abstraction	0 – 65.7 (69/105)	2 – 70.5% (60/85)	4 – 72.7 (8/11)

### Analyse bivariée

Cours_db (Oui) – 90%	Autodétermination (2) – 99%
----------------------	-----------------------------

### DR

Sexe (M) – 90%	Autodétermination (2) – 99%
----------------	-----------------------------

### AIC

Sexe (M) – 90%	Autodétermination (2) – 99%
----------------	-----------------------------

### Arbre de décision

Autodétermination = 2 → 4 réussites	Statut = 0 → 4 réussites, 1 échec
Autodétermination = 1 → 3 réussites, 1 échec	Statut = 1 → 3 réussites, 1 échec
Autodétermination = 0 → 3 réussites, 2 échecs	Statut = 2 → 3 réussites, 1 échec
	<b>Statut = 3 → 2 réussites</b>
Sexe = M → 3 réussites, 1 échec	<b>Cours_algo = Oui → 1 échec</b>
<b>Sexe = F → 3 réussites</b>	Cours_algo = Non → 3 réussites, 1 échec
Abstraction = 4 → 1 réussite	Langage = 0 → 1 réussite
Abstraction = 2 → 1 réussite, 1 échec	Langage = 1 → 1 réussite
<b>Abstraction = 0 → 1 réussite</b>	



### VII.3.2. Profil

De tous les résultats présentés pour la population INGMI, il peut être déduit que :

- Le sexe est un facteur important influençant la réussite du cours d'introduction à la programmation surtout s'il est féminin. Être fille en INGMI influence la réussite qu'être garçon.
- Les étudiants en situation d'étalement ou provenant d'autres universités ont une certaine influence à la réussite, avec un taux de réussite à 100%.
- Aucune variable du bagage informatique n'a été retenu comme facteur de réussite. D'ailleurs, la plupart des variables du bagage informatique ont un taux de réussite élevée lorsque l'étudiant a répondu « Non » pour un des cours, sauf pour le cours de création de pages Web où le taux de réussite pour les « Oui » est plus élevé que les « Non ».
- Même constat pour les variables du bagage en programmation. Le fait d'avoir étudié la programmation au secondaire n'a aucune influence sur la réussite et le fait d'avoir étudié le cours d'algorithme au secondaire paraît même comme un facteur d'échec lorsqu'associé avec une autodétermination non élevée. Cela peut se voir aussi, lorsqu'on regarde son taux de réussite de 0%. Aucun des 3 élèves ayant eu le cours d'algorithme a réussi le cours d'introduction à la programmation. Le fait d'avoir appris au moins un langage de programmation n'influence pas non plus le résultat bien que le taux de réussite soit supérieur à 70%.
- Le fait d'avoir appris le langage de programmation Python n'a aucune influence sur le résultat.
- Concernant la motivation, il apparaît clairement qu'une autodétermination élevée est un fort facteur de réussite pour l'étudiant d'INGMI. Dans toutes les méthodes d'analyse utilisées pour ce mémoire, l'autodétermination a toujours été fortement significative.

## VII.3. Profil des étudiants en INFO

### VII.3.1. Rappel des résultats

#### Taux de réussite

66.7% (192/288)

#### Taux de réussite par sexe

Filles – 53.6% (15/28)

Garçons – 68.5% (178/260)

### Taux de réussite par statut

Nouveaux – 67.3% (165/245)  
Autres filières – 64% (16/25)

Bisseurs – 50% (4/8)  
Autres cas – 80% (8/10)

### Taux de réussite par variables du bagage en informatique

Cours\_info\_sec : Oui – 68.5% (63/92)  
Cours\_info\_sec : Non – 66.3% (130/196)

Bg\_info\_56 : **Oui – 74% (54/73)**  
Bg\_info\_56 : Non – 64.6% (139/215)

Cours\_codage\_info : **Oui – 75% (45/60)**  
Cours\_codage\_info : Non – 64.9% (148/228)

Cours\_ordi\_arch : Oui – 68.6% (48/70)  
Cours\_ordi\_arch : Non – 66.5% (145/218)

Cours\_os : Oui – 67.2% (41/61)  
Cours\_os : Non – 67% (152/227)

Cours\_reseau : **Oui – 70.2% (33/47)**  
Cours\_reseau : Non – 66.4% (160/241)

Cours\_html : **Oui – 73.6% (53/72)**  
Cours\_html : Non – 64.8% (140/216)

Cours\_db : **Oui – 75.5% (37/49)**  
Cours\_db : Non – 65.3% (156/239)

### Taux de réussite par variables du bagage en programmation

Cours\_programmation  
Cours\_algo  
Langage

Oui – **76.8% (53/69)**  
Oui – **76.4% (42/55)**  
1 – 77.5 (124/160)

Non – 63.9% (140/219)  
Non – 64.8% (151/233)  
0 – 53.9% (69/128)

### Taux de réussite par variables du bagage en Python

Python

1 – **76.3% (58/76)**

0 – 63.7% (135/212)

### Taux de réussite par variables liées à la motivation

Raison\_filière

Intérêt – 67% (103/153)  
Passion – 65% (30/46)

Facilité – **75.7% (28/37)**  
Programmation – **71.7% (43/60)**

Autodidacte\_prog : **Oui – 82.1% (78/95)**  
Autodidacte\_prog : **Non – 59.6% (115/193)**

Cours\_prepa : **Oui – 75.7% (84/111)**  
Cours\_prepa : Non – 61.6% (109/177)

Autodétermination  
Abstraction

**0 – 55.6% (10/18)**  
0 – 68 (85/125)

1 – 65.2% (88/135)  
2 – 67.7% (88/130)

**2 – 70.4% (95/135)**  
4 – 60.6 (20/33)

### Analyse bivariable

Cours\_programmation (Oui) – 95%  
Autodidacte\_prog (Oui) – 99.9%

Langage (1) – 99.9%  
Cours\_prepa (Oui) – 95%

Python (1) – 95%

### DR

Autodidacte\_prog (Oui) – 90%

Langage (1) – 90%

Cours\_prepa (Oui) – 95%

### AIC

Autodidacte\_prog (Oui) – 90%  
Cours\_info\_sec (Oui) – 95%

Langage (1) – 90%  
Cours\_programmation (Oui) – 90%

Cours\_prepa (Oui) – 99%

### Arbre de décision

Langage = 1 → 3 réussites, 1 échec Langage = 0 → 2 réussites, 2 échecs	Cours_prepa = Oui → 2 réussites Cours_prepa = Non → 3 réussites, 2 échecs
Statut = 0 → 2 réussites Statut = 1 → 1 réussite, 1 échec Statut = 2 → 2 réussites, 1 échec Statut = 3 → 2 réussites, 1 échec	Cours_codage_info = Oui → 1 échec Cours_codage_info = Non → 2 réussites, 1 échec
Cours_info_sec = 1 → 1 réussite, 1 échec	Autodétermination = 0 → 1 réussite Autodétermination = 1 → 1 réussite Autodétermination = 2 → 1 réussite
Abstraction = 0 → 1 réussite Abstraction = 2 → 1 réussite	Cours_os = Oui → 1 réussite

### VII.3.2. Profil

De tous les résultats présentés pour la population INFO, il peut être déduit que :

- Le statut de l'étudiant a une certaine influence sur le résultat en jetant un coup d'œil sur l'arbre de décision. En effet, le fait d'être un étudiant nouveau ou un étudiant venant d'une autre filière, combiné au fait d'avoir déjà appris un langage de programmation influencent la réussite.
- Au niveau du **bagage en informatique**, avoir eu un cours d'informatique au secondaire est statistiquement significatif pour les modèles des méthodes utilisées et auraient une influence sur la réussite si combinée avec une participation au cours préparatoire.
- Au niveau du **bagage en programmation**, le fait d'avoir appris un langage de programmation influence la réussite dépendamment du statut de l'étudiant qui devrait être nouveau ou venant d'une autre filière. Le fait d'avoir eu le cours de programmation influence la réussite (méthode AIC) bien que n'apparaissant pas dans l'arbre de décision dû au fait que la profondeur a été limitée à 6 pour la population INFO.
- Le fait d'avoir appris le langage Python montre un taux de réussite élevé pour les étudiants d'INFO (76.3%) mais aucune analyse multivariée ne nous indique une certaine influence de cette variable par rapport à la réussite.
- Pour les variables liées à la motivation, les résultats montrent que pour les variables liées au fait de participer aux cours préparatoires a une influence significative sur la réussite au cours d'introduction à la programmation. Le fait d'avoir appris seul au moins un langage de programmation aurait une influence sur la réussite bien que n'étant pas présent dans l'arbre de décision à cause de la profondeur limitée volontairement à six.

## VIII. Discussion

*« Est-ce que les étudiants ayant eu des cours dont les compétences sont reconnues comme développant la pensée computationnelle en secondaire, ont-ils mieux réussi le cours d'introduction à la programmation que ceux n'en ayant pas eu ? »*

Avant de répondre à cette hypothèse, il est important de rappeler les cours qui ont été considérés comme développant la pensée computationnelle :

- Codage de l'information
- Architecture de base d'un ordinateur
- Réseau
- Programmation
- Réalisation d'algorithme
- Bases de données

En ce qui concerne **les étudiants d'INGMI**, il est apparu que cette hypothèse est fausse. En effet, en regardant les résultats présentés au point VII.3, montrent statistiquement un taux de réussite élevé pour les étudiants n'ayant pas eu ces cours à l'école secondaire. Le nombre d'étudiants ayant eu ces cours étant très bas (inférieur à 5% de la population INGMI), nul ne sait ce que ça aurait pu donner pour un nombre d'étudiants plus élevé.

En ce qui concerne **les étudiants d'INFO**, il est apparu que cette hypothèse est vérifiée. En effet, les résultats présentés au point VII.4, montrent statistiquement un taux de réussite supérieur pour les étudiants ayant eu ces cours à l'école secondaire.

*« Est-ce que les étudiants ayant suivi des cours d'algorithme et de programmation en secondaire sont plus susceptibles de réussir le cours d'introduction à la programmation que les autres n'en ayant pas eu ? »*

Cette hypothèse a été prouvée en répondant à la première hypothèse puisque les cours de programmation et d'algorithme font partie des cours développant la pensée computationnelle.

*« Est-ce les étudiants ayant déjà rencontré le langage Python ont plus de chance de réussir le cours de Programmation que les autres ne l'ayant pas rencontré. »*

Pour les **étudiants d'INGMI**, la réponse est un petit « oui ». En effet, le taux de réussite pour les étudiants ayant déjà appris le langage Python est de peu supérieur à celui des étudiants ne l'ayant pas appris et il apparaît encore le même problème de l'échantillon (13/201) représentant les étudiants ayant déjà appris le langage Python. Cela ne représente même pas 10% de la population INGMI. Même si le taux est supérieur, avec ce peu nombre d'étudiants ayant appris le langage Python, il est difficile de répondre par « Oui » ferme.

Concernant la **population INFO**, l'échantillon de ceux qui ont déjà rencontré le langage Python est suffisamment grand (76 sur 288, soit 26% de la population INFO) pour considérer la réponse comme étant significative. La réponse à cette hypothèse est « Oui » puisque le taux de réussite (76.3%) est supérieur par rapport aux étudiants ne l'ayant pas appris (63.7%).

*« Est-ce que les étudiants qui n'ont pas suffisamment de motivation vis-à-vis de l'apprentissage de la programmation sont moins susceptibles de réussir le cours d'introduction à la programmation que ceux qui ont une motivation à apprendre à programmer. »*

Il est important de définir d'abord ce qu'est une motivation suffisante. En effet, en énonçant cette hypothèse, le but était donc d'évaluer les raisons du choix de filière informatique données par les étudiants. Tout au long de cette étude, deux variables avaient été créées pour mesurer cette motivation. Bref, répondre à cette revient à regarder au taux de réussite des étudiants au niveau de l'autodétermination et du niveau d'abstraction.

Pour les **étudiants d'INGMI**, le taux de réussite au niveau de l'autodétermination montre que plus elle augmente, plus le taux de réussite augmente (0 → 52%, 1 → 65%, 2 → 88%). Même chose pour le niveau d'abstraction (0 → 65.7%, 2 → 70.5%, 4 → 72.7%), même si le nombre d'étudiants de niveau d'abstraction élevé ne représente que 5.5% de la population INGMI.

Pour les **étudiants d'INFO**, cela n'est le cas que pour l'autodétermination où il apparaît que plus on monte au seuil de l'autodétermination, plus on a des chances de réussir (0 → 55.6%, 1 → 65.2%, 2 → 70.4%) au cours d'introduction à la programmation contrairement au niveau d'abstraction pour lequel plus on monte en niveau d'abstraction, moins on a des chances de réussir (0 → 68%, 2 → 67.7%, 4 → 60.6%)

## VIII.1. Biais

L'objectif de ce point est de montrer que certains résultats obtenus durant l'analyse des données issues de l'enquête soumise aux étudiants *pourraient* être trompeurs. Plusieurs autres aspects, susceptibles d'influencer les résultats des étudiants au cours d'introduction à la programmation, n'ont pas été évoqués, ni évalués dans ce mémoire.

Concernant par exemple la **population INGMI**, les résultats présentés au point VII.3.1 montrent que tous les étudiants (3) qui ont eu un *cours d'algorithme* ont échoué au cours d'introduction à la programmation comme la moitié des huit étudiants ayant eu des cours de programmation alors que ces deux matières comprennent fortement les éléments de la pensée computationnelle. Après vérification, il s'avère que les trois étudiants ayant eu le cours d'algorithme en secondaire font partie des quatre étudiants ayant eu le cours de programmation en secondaire et qui ont échoué au cours d'introduction à la programmation. Voici quelques raisons pouvant expliquer ce genre de résultat :

- **Nombre d'étudiants**

Les nombres d'étudiants ayant eu ces cours en secondaire (trois pour le cours d'algorithme et huit pour le cours de programmation) pour la population INGMI sont très petits pour tirer une conclusion ferme. Nul ne sait si cela aurait été le cas si les nombres d'étudiants étaient élevés

- **INGMI plutôt qu'INFO**

Le personnel enseignant dans la faculté INGMI de l'université de Namur explique que beaucoup d'étudiants, bien qu'ayant appris un cours de programmation ou d'algorithme en secondaire, préfèrent s'inscrire en INGMI afin d'éviter des cours de programmation en INFO, ne sachant pas qu'il y a un cours d'introduction à la programmation donnée en INGMI. Beaucoup d'entre eux ne savent pas qu'il y a un cours d'introduction à la programmation.

D'autres raisons expliquant les résultats au cours d'introduction à la programmation et concernant les étudiants des deux facultés :

- **Peu de connaissances sur les cours**

Lors de leur choix, peu d'étudiants savent exactement ce qui se cache derrière les cours faisant partie de leur filière. D'autres n'auraient pas bien compris les intitulés des cours ni leur exact contenu. Concernant le cours d'introduction à la programmation, la méconnaissance des étudiants de ce cours a été aussi constaté par le personnel enseignant.

- **Désirabilité sociale**

Il y a des étudiants qui ont répondu aux questions de l'enquête sous le regard de l'assistante, ce qui a pu générer des réponses qui les présentent sous un jour favorable inconsciemment ou consciemment afin de montrer une bonne image d'eux-mêmes.

- **Obligation de participation**

Les étudiants ont dû se sentir obligés de répondre à cette enquête et de bien le faire parce qu'étant en face de leur évaluateur. Ce qui a pu impacter aussi leur réponse et leur manière de répondre.

- **Raisons socio-économiques et culturelles ?**

Les données collectées via l'enquête soumise aux étudiants ne permettent pas d'évaluer les facteurs socio-économiques. Ces derniers peuvent influencer les notes obtenues par un étudiant [Galand, B., Frenay, M. & Bourgeois, E., 2004].

## IX. Conclusion

Pour cette étude, tout reposait sur l'enquête soumise aux étudiants. Les réponses à cette enquête constituaient la base de données à exploiter et à utiliser pour répondre à la question de recherche liée au sujet de mémoire « profiler les étudiants pour leur donner envie de réussir ». Le nombre total des étudiants pour lesquels les réponses étaient utilisables étaient au nombre de 489 : 201 étudiants d'INGMI et 288 étudiants d'INFO.

Tout a commencé par l'analyse des questions posées. Le but était de regarder quelles informations pourraient être extraites et intéressantes pour traiter le sujet. En effet, le questionnaire était fait de telle sorte à ressortir les compétences qu'auraient déjà acquises les étudiants avant le début de l'année académique. La question de recherche a été donc basée sur cet objectif là c'est-à-dire de profiler les étudiants au niveau de leur bagage en informatique, leur bagage en programmation, leur connaissance du langage de programmation Python et aussi déceler leur niveau de motivation, leurs objectifs. Le but était donc de trouver parmi les réponses données par les étudiants, les facteurs qui influenceraient la réussite ou l'échec des étudiants.

Pour répondre à cette problématique, les données ont été soumises à des traitements afin de faciliter la méthodologie de travail. Parmi ces transformations, les raisons du choix de filière informatique évoquées par les étudiants. Les réponses liées à cette question ont fait l'objet d'une analyse lexicale dans le but de catégoriser ces raisons pour une meilleure analyse. Dans ce processus, une part de subjectivité est intervenue au moment où il fallait déterminer pour une raison donnée, la catégorie à laquelle elle appartenait et ceci en parcourant toutes les réponses liées à cette question. Les raisons données ont permis aussi d'extraire le niveau d'autodétermination et le niveau d'abstraction auxquels appartient chaque raison évoquée. Cet exercice, détaillé dans ce mémoire, a demandé beaucoup d'interprétation personnelle tant pour déterminer le seuil d'autodétermination que pour décider du niveau d'abstraction de la raison donnée. Ces interprétations personnelles auraient pu subir plusieurs contrôles par des personnes tierces pour confirmer le résultat ou émettre un avis contraire. Elles pouvaient être le résultat d'une discussion ou d'une concertation entre plusieurs personnes mais cela n'a pas été le cas. D'autres transformations étaient plus directes et objectives.



Pour la méthodologie du travail, le choix était porté sur la régression logistique et l'arbre de décision. Une *analyse bivariée* basée sur la régression logistique a été effectuée pour chaque variable des deux populations INGMI et INFO avec la variable représentant la note des étudiants du cours d'introduction à la programmation. Cette analyse a permis de voir que l'autodétermination élevée des étudiants d'INGMI était statistiquement très significative à 99% et le fait d'avoir eu des cours de bases de données aussi statistiquement significatif à 90%. Pour la population INFO, le fait d'avoir eu le cours de programmation, d'algorithme, le fait d'avoir appris au moins un ou plusieurs langages de programmation et de l'avoir appris seul, parmi lesquels le langage Python sont apparus être statistiquement significatifs pour cette analyse.

Ensuite une *analyse multivariée*, dans laquelle toutes les variables ont été évaluées ensemble afin de déterminer lesquelles influencent le modèle. Le but était de déterminer les facteurs, les variables qui influencent la réussite ou l'échec des étudiants au cours d'introduction à la programmation en ne gardant que dans le modèle les variables importantes. Pour cela, deux méthodes ont été utilisées : Une première méthode utilisant l'évaluation par la déviance résiduelle « DR ». Cette méthode a permis de montrer que l'autodétermination élevée avait une influence sur le modèle de la population INGMI ainsi que le sexe. Pour la population INFO, le fait d'avoir appris un ou plusieurs langages de programmation et de l'avoir fait seul, ainsi la participation aux cours préparatoires sont apparus être statistiquement significatifs pour le modèle.

L'autre méthode était basée sur le critère d'information d'Akaike « AIC » dont le résultat pour la population INGMI est resté le même que pour la méthode DR. Pour la population INFO, deux autres variables se sont ajoutées au résultat de la méthode DR : le fait d'avoir eu des cours d'informatique en secondaire et le fait d'avoir eu le cours de programmation en secondaire.

Avec l'arbre de décision, le but était de définir les variables aboutissant à une réussite ou un échec à plus de 80%. Les résultats ont confirmé pour la population INGMI que l'autodétermination élevée et le fait d'être une fille influencent fortement la réussite. Pour la population INFO, le fait d'avoir eu des cours d'informatiques en secondaire combinée avec la participation au cours préparatoire influencent la réussite des étudiants. À cela, s'ajoutent aussi, le fait d'avoir appris un langage de programmation pour les nouveaux étudiants ou ceux qui viennent d'une autre filière.

Cependant, pour arriver à ces résultats, plusieurs soucis ont été rencontrés durant la préparation de ce mémoire. Le questionnaire ne permettant pas d'avoir toutes les informations nécessaires telles que le sexe des étudiants et la note du cours d'introduction à la programmation, nous étions donc confrontés au règlement général sur la protection des données (RGPD) et nous devions avoir l'accord du DPO (« Data Protection Officer ») pour entrer en possession de certaines informations liées aux données et pouvoir les traiter. Cette partie a donc pris plus de temps et a réduit le temps d'analyse.

En résumé, pour ce mémoire, une des méthodes d'analyse aurait été d'utiliser des méthodes de prédiction de machine learning en séparant les données en deux groupes : un groupe de données pour apprendre et un autre groupe pour tester le modèle trouvé. Cette option était la première mais a été abandonnée car le nombre d'étudiants par population (201 pour INGMI et 288 INFO) était jugé pas suffisant pour avoir plus de précision et plus de données sur lesquelles apprendre et tester.

## X. Bibliographie

- [1] Administration générale de l'Enseignement, "Programme d'études INFORMATIQUE", <http://www.wallonie-bruxelles-enseignement.be/progr/477-2016-248a.pdf>, Belgique Francophone, 2016
- [2] Adu-Manu, Kofi & Arthur, John & Amoako, Prince. (2013). Causes of Failure of Students in Computer Programming Courses: The Teacher Learner Perspective. *International Journal of Computer Applications*. 77. 27-32. 10.5120/13448-1311.
- [3] Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, 55, 832–835.
- [4] Amnah Alshahrani, Isla Ross, and Murray Wood. 2018. « Using social cognitive career theory to understand why students choose to study computer science ». In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ACM, 205–214.
- [5] Andrzejewska, Magdalena. (2018). Difficulties in learning introductory programming - student's perspective.
- [6] Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to “Real” programming. *ACM Transactions on Computing Education*, 14(4), 25.
- [7] B. Hanks, C. McDowell, D. Draper and M. Krnjajic. Program quality with pair programming in CS1. In *ITiCSE '04: Proceedings of the 9<sup>th</sup> annual SIGCSE conference on Innovation and technology in computer science education* pages 176-180, Leeds, United Kingdom, 2004.
- [8] Bergin, Susan & Reilly, Ronan. (2005). Programming: Factors that Influence Success. *ACM Sigcse Bulletin*. 37. 411-415. 10.1145/1047124.1047480
- [9] Boudrenghien, Gentiane & Frenay, Mariane & Bourgeois, Etienne, "La transition de l'enseignement secondaire vers l'enseignement supérieur : rôle des représentations et motivations de l'étudiant à l'égard de son projet de formation", in "Persévérer et réussir à l'Université", UCL Presses Universitaires de Louvain, pages 55-82, 2013
- [10] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Annual American Educational Research Association meeting*, Vancouver, BC, Canada.
- [11] Brunstein, J. C. (1993). Personal goals and subjective well-being: A longitudinal study. *Journal of Personality and Social Psychology*, 65(5), 1061-1070.
- [12] C. Bereiter and E. Ng. “Three Levels of Goal Orientation in Learning”, *Journal of the Learning Sciences*, vol. 1, n° 3, pp. 243-271, 1991.
- [13] Carver, C. S., & Scheier, M. F. (1998). *On the self-regulation of behavior*. Cambridge: Cambridge University Press.

- [14] D. Nandigam and H. Bathula, "Competing Dichotomies in Teaching Computer Programming to Beginner-Students," pp. 1-9, December 2011.
- [15] Deci, E. L. et Ryan, R. M. (1985). The General Causality Orientations Scale: Self-Determination in Personality. *Journal of Research in Personality*, 19, 109-134.
- [16] Deci, E. L. et Ryan, R. M. (2002). *Handbook of Self-Determination Research*. Rochester, NY : University of Rochester Press.
- [17] Dupriez, V. & Vandenberghe, V. (septembre 2003). L'école en Communauté française de Belgique : De quelle inégalité parlons-nous? Communication orale au congrès de l'Admée, Liège, Belgique.
- [18] Dunican, E. (2002). Making the analogy: Alternative delivery techniques for first year programming courses. In J. Kuljis, L. Baldwin & R. Scoble (Eds), *Proceedings from the 14th Workshop of the Psychology of Programming Interest Group*, Brunel University, June 2002, 89-99.
- [19] Galand, Benoît; Neuville, Sandrine; Frenay, Mariane. L'échec à l'université en Communauté française de Belgique : comprendre pour mieux prévenir ?. In: *Cahiers de Recherche en Education et Formation*, Vol.39, p. 5-17 (2005) <http://hdl.handle.net/2078.1/93664>
- [20] Galand, B., Frenay, M. & Bourgeois, E. (2004). Facteurs de réussite en 1<sup>o</sup> candidature à la Faculté des Sciences et à l'Institut d'Education Physique et de Réadaptation. Communication orale à la Journée d'étude de la Chaire de pédagogie universitaire, Louvain-la-Neuve, Belgique.
- [21] GANASSALI S. (2014), *Enquête et analyse de données avec Sphinx*, Pearson.
- [22] Germeijs, V., & Verschueren, K. (2007). High school students' career decision-making process: Consequences for choice implementation in higher education. *Journal of Educational Behavior*, 70, 223-241.
- [23] Gomes, Anabela & Mendes, Antonio. (2007). Learning to program - difficulties and solutions. 283-287
- [24] Grover, Shuchi & Pea, Roy. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*. 42. 38-43.
- [25] Hagan, D. and Markham, S. 2004. Does it help to have some programming experience before beginning a computing degree program? *ACM SIGCSE Bulletin*. 32, 3 (2004), 25–28.
- [26] Henry, J & Joris, N 2016, Informatics at secondary schools in the French-speaking region of Belgium: myth or reality? In \*. *The International Conference on Informatics in Schools: Situation, Evolution and Perspectives*, Münster, Germany, 13/10/16.

- [27] Holden, E. and Weeden, E. 2004. The impact of prior experience in an information technology programming course sequence. (2004), 41.
- [28] Hollenbeck, J. R., & Klein, H. J. (1987). Goal commitment and the goal-setting process: Problems, prospects, and proposals for future research. *Journal of Applied Psychology*, 72, 212-220.
- [29] Kaczmarczyk, L., & Dopplick, R. (2014). Rebooting the pathway to success: Preparing students for computing workforce needs in the United States. Education Policy Committee, Association for Computing Machinery.
- [30] Kirsti Ala-Mutka, "Problems in learning and teaching programming," a literature study for developing visualizations in the Codewitz-Minerva project, pp. 1 – 13, (n.d)
- [31] Külli Kori, Margus Pedaste, Heilo Altin, Eno Tõnisson, and Tauno Palts. 2016. Factors That Influence Students' Motivation to Start and to Continue Studying Information Technology in Estonia. *IEEE Transactions on Education* 59, 4 (2016), 255–262
- [32] Külli Kori, Margus Pedaste, Margus Niitsoo, Rein Kuusik, Heilo Altin, Eno Tõnisson, Inga Vau, Äli Leijen, Mario Mäeots, Leo Siiman, et al. 2014. Why do students choose to study Information and Communications Technology? *The European Procedia Social and Behavioral Sciences* 191 (2014), 2867–2872.
- [33] M. Butler and M. Morgan, "Learning challenges faced by novice programming students studying high level and low feedback concepts," *Proceedings ascilite Singapore*, pp. 99 – 107, 2007.
- [34] M. Bruinsma, "Motivation, cognitive processing and achievement in higher education," *Learning and Instruction*, vol. 14, n°6, pp. 549-568, 2004.
- [35] M. N. ISMAIL, N. A. NGAH and I. N. UMAR, "Instructional strategy in the teaching of computer programming: a need assessment analyses," *TOJET: The Turkish Online Journal of Educational Technology*, volume 9 Issue 2, pp. 569–571, April 2010.
- [36] N. Entwistle, "Concepts and conceptual frameworks underpinning the ETL project," *Occasional Report 3*, Edinburgh, Scotland: ETL Project, 2003.
- [37] Nowaczyk, R. 2019. Cognitive skills needed in computer programming paper presented at the annual meeting of the southeastern psychological association (March. (2019).
- [38] Päivi Kinnunen, Matthew Butler, Michael Morgan, Aletta Nysten, Anne-Kathrin Peters, Jane Sinclair, Sara Kalvala, and Erkki Pesonen. 2016. Understanding initial undergraduate expectations and identity in computing studies. *European Journal of Engineering Education* 43, 2 (2016), 201–218.
- [39] Patil, S.P. and Goje, A.C. 2009. The effect of developments in student attributes on success in programming of management students. 2009 International Conference on Education Technology and Computer, ICETC 2009. (2009), 191–193.

- [40] Quinn, H., Schweingruber, H., & Keller, T. (Eds.). (2012). A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. National Academies Press.
- [41] J. Vallerand and R. Bissonnette, "On the predictive effect of intrinsic, extrinsic, and amotivation styles on behaviour: A prospective study," *Journal of Personality*, vol. 60, pp. 599-620, 1992.
- [42] Ronan. (2005). Programming: Factors that Influence Success. *ACM Sigcse Bulletin*. 37. 411-415. 10.1145/1047124.1047480.
- [43] Säde, M., Suviste, R., Luik, P., Tõnisson, E., & Lepp, M. (2019). Factors That Influence Students' Motivation and Perception of Studying Computer Science. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*
- [44] Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. In J. Carter, I. Utting, & A. Clear (Eds.), *Proceedings of 18th annual conference on innovation and technology in computer science education* (p. 6). Canterbury: University of Southampton.
- [45] Sury, C., "En 2020, le candidat à la recherche d'un emploi restera roi", <https://www.lecho.be/monargent/travail/en-2020-le-candidat-a-la-recherche-d-un-emploi-restera-roi/10195793.html>, 06/01/2020
- [46] TeresaStat (<https://stats.stackexchange.com/users/54587/teresastat>), Interpreting Residual and Null Deviance in GLM R, URL (version: 2017-01-28): <https://stats.stackexchange.com/q/113022>
- [47] T. Jenkins, "On the difficulty of learning to program," in *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, vol. 4, 2002, pp. 53–58.
- [48] Thorson K, « Early Learning Strategies for Developing Computational Thinking Skills »  
<https://www.gettingsmart.com/2018/03/early-learning-strategies-for-developing-computational-thinking-skills/>, (18/03/2018)
- [49] US Bureau of Labor Statistics. Computer and information technology occupations, 2013.
- [50] Watson, Christopher and Li, Frederick W. B. (2014) 'Failure rates in introductory programming revisited.' in *Proceedings of the 2014 conference on Innovation technology in computer science education (ITiCSE '14)*. New York: Association for Computing Machinery (ACM), pp. 39-44.
- [51] Werth, L.H. 1986. Predicting Student Performance in a Beginning Computer Science Class. *SIGCSE Bull.* 18, 1 (1986), 138–143.

- [52] Wiedenbeck, S. et al. 2004. Factors affecting course outcomes in introductory programming. 16<sup>th</sup> Workshop of the Psychology of Programming Interest Group. April (2004), 97–110.
- [53] Wing, J. E. (2011). Research notebook: Computational thinking-What and why? The Link Magazine, spring. Carnegie Mellon University, Pittsburgh.
- [54] Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in scienceclassrooms. Communications of the ACM, 57(8), 24–28.
- [55] Wilson, B.C. and Shrock, S. 2004. Contributing to success in an introductory computer science course. ACM SIGCSE Bulletin. 33, 1 (2004), 184–188.

## XI. Annexe

### XI.1. Questionnaire soumis aux étudiants

**Les réponses aux questions avec « \* », sont obligatoires.**

**Question 1 \***

Mon eID à l'UNamur.

**Question 2 \***

Je suis ...

- ☐ En INGMI 1-3
- ☐ En INGMI 2-4
- ☐ En INFOA
- ☐ En INFOB

**Question 3 \***

J'effectue actuellement ma première année à l'Université.

Répondre "Non" si vous bissez (échec), si vous allégez (étalement) ou si vous vous réorientez (changement de filière).

- Oui
- Non, je suis en situation d'étalement
- Non, je bisse ma première année
- Non, j'ai effectué une première année dans une autre filière
- Autre

**Question 4**

Si vous avez répondu "Non" : Je cite le(s) langage(s) de programmation que j'ai déjà appris à l'Université.

**Question 5 \***

J'ai choisi la filière informatique (études d'informatique OU option informatique en INGMI) parce que... (J'explique ici, en quelques phrases, les raisons de mon choix.)

**Question 6 \***

J'ai choisi l'UNamur parce que...

**Question 7 \***

Pour moi, l'informatique c'est ...

**Question 8 \***

Pour moi, un ordinateur c'est ...

**Question 9 \***



De quoi est capable un ordinateur ? Qu'est-ce qui est à sa portée ?

**Question 10 \***

Qu'est-ce qui ne sera jamais à la portée d'un ordinateur ?

**Question 11 \***

J'ai eu des cours d'informatique durant mes études secondaires.

- ☐ Oui
- ☐ Non

**Question 12 \***

J'ai fait mes études secondaires (de la 1<sup>re</sup> à la 6<sup>e</sup> année) en Communauté française de Belgique.  
Cocher "Autre" si vous avez effectué une partie de vos études ailleurs qu'en Communauté française de Belgique. Précisez alors quelle(s) année(s) et où.

- ☐ Oui
- ☐ Non

**Question 13 \***

Je précise en quelle(s) année(s) j'ai eu un cours d'informatique.

	J'ai eu cours d'info	Je n'ai pas eu cours d'info	Je ne m'en souviens pas
1 <sup>ère</sup> secondaire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2 <sup>ème</sup> secondaire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3 <sup>ème</sup> secondaire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4 <sup>ème</sup> secondaire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5 <sup>ème</sup> secondaire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6 <sup>ème</sup> secondaire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Question 14 \***

Dans mon (mes) cours d'informatique, je me souviens avoir abordé les notions suivantes :

	Je suis certain que oui	Je suis certain que non	Je pense que oui	Je pense que non	Je ne me souviens pas
Le codage de l'information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
L'architecture de base d'un ordinateur (composants) et de son environnement matériel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Le système  
d'exploitation  
(fonctions et  
rôles) et la  
gestion de  
fichiers

☐ ☐ ☐ ☐ ☐

Les réseaux et  
Internet

☐ ☐ ☐ ☐ ☐

La création de  
page Web  
(langage  
HTML)

☐ ☐ ☐ ☐ ☐

La réalisation  
d'algorithmes

☐ ☐ ☐ ☐ ☐

La  
programmation

☐ ☐ ☐ ☐ ☐

La création de  
base de  
données

☐ ☐ ☐ ☐ ☐

### Question 15

Le cas échéant, je cite le(s) langage(s) de programmation que j'ai apprise durant mes secondaires

### Question 16 \*

J'ai appris par moi-même certains langages de programmation.

- ☐ Oui
- ☐ Non

### Question 17

Le cas échéant, je cite ici le(s) langage(s) de programmation auxquels je me suis initié

### Question 18 \*

J'ai participé aux cours préparatoires.

- ☐ Oui
- ☐ Non

### Question 19

Les séances de programmation avec micro bit vous ont-elles aidé à comprendre certaines notions d'informatique ? Si oui, lesquelles. Si non, expliquez pourquoi.

**Question 20**

Si vous aviez le choix du contenu des cours préparatoires (en ce qui concerne l'informatique et la programmation), que souhaiteriez-vous y faire ?